



**Escola Politècnica Superior
de Castelldefels**

UNIVERSITAT POLITÈCNICA DE CATALUNYA

TREBALL DE FI DE CARRERA

TÍTOL DEL TFC: Implantación de una centralita VoIP basada en Asterisk en un parque de innovación

TITULACIÓ: Enginyeria Tècnica de Telecomunicació, especialitat Telemàtica

AUTOR: Jordi Piñol Casas

DIRECTOR extern: Javier Fernandez Garcia

DIRECTOR: Juan López Rubio

DATA: 5 de Gener del 2011

Título: Implantación de una centralita VoIP basada en Asterisk en un parque de innovación

Autor: Jordi Piñol Casas

Director: Javier Fernández García / Juan López Rubio

Data: 5 de Enero del 2011

Resumen

Este TFC nace como parte de un proyecto de Esade para proveer de servicios IT un parque de nueva creación. Concretamente la intención es encontrar una solución técnica al servicio de telefonía de un parque de negocios.

El parque de negocios en cuestión es ESADECREAPOLIS. Se define como un parque de la innovación donde empresas de múltiples sectores se juntan en un entorno compartido con el objetivo común de acelerar su innovación de una manera abierta y cruzada.

Con la definición del parque, se pueden intuir varios de los objetivos que tendrá el proyecto. En primer lugar, la necesidad de implementar una solución de alto rendimiento para intensificar la comunicación entre residentes está implícita en la definición de la empresa. Los residentes tienen que tener canales directos para comunicarse entre ellos y no pueden notar la pérdida de funcionalidades respecto a sus sedes. En segundo lugar, la delicada situación que atraviesan, en general, las empresas implica que la solución ofertada tenga que ser de bajo coste.

Para conseguir estos objetivos hemos apostado por la Voz sobre IP como base de trabajo. El avance de la tecnología VoIP en los últimos años es incuestionable, ha dejado de ser una tecnología emergente para consolidarse como la única opción viable para una inversión de esta índole. En el entorno empresarial, consolidada en gran parte de la mano de productos cerrados. En nuestro caso, daremos un giro y utilizaremos la centralita telefónica Asterisk, que funcionará con un sistema Unix FreeBSD como sistema operativo.

El proyecto se divide en dos partes diferenciadas, en primer lugar tenemos una breve descripción del modelo sobre el que configuraré el servicio en el parque y después comentaré la configuración sobre la que trabajaremos para implantar el servicio.

Title: Asterisk VoIP Private Branch Exchange Implementation in innovation park

Author: Jordi Piñol Casas

Director: Javier Fernández García / Juan López Rubio

Date: January, 05th 2011

Overview

This TFC was created as part of an Esade project to provide IT services to a new business park. The main objective is to find a good, optimal and satisfactory technical solution to the phone communications.

This business park is called ESADECREAPOLIS. It can be described as an innovation park where different companies from different areas are there to share the environment, stimulate each other's ideas and at the end of the day to reach a common goal which is accelerating the innovation's process.

The more we know how this business park works the more we are able to get an idea about which will be the main targets of this project. First of all we can see there is a need to implement a high performance solution to enhance communication between the members of this community, this is implicit in the definition of business. The different companies must be able to communicate with the others in a direct and easy way, the rapprochement between them has to be a reality, technology has to make things easier for them and can not be an issue or excuse for them not to develop correctly in this environment, they can not notice the lag of functionality in their companies because of that. Second of all, we have to be realistic in the way of facing the difficult economical situation we have at the moment and the solution offered has to be low cost.

To achieve these targets we have decided to use IP phones, since they take full advantage of converged voice and data networks. The IT advances in this research field for the last few years have been obviously huge, and is not longer an emerging technology but the only factible option for such an investment. In the business environment, consolidated much of the labor unopened products. We have decided to work with the switchboard Asterisk and it will be running on a Unix FreeBSD OS.

This project has two different parts, first of all I will briefly describe the model that will be configured to provide VoIP service to the business park and after I will do a summary about which settings I will be using to implement this service.

ÍNDICE

ÍNDICE	5
INTRODUCCIÓN	7
CAPÍTULO 1. MODELO INICIAL	9
1.1. Esadecreapolis	9
1.1.1. ¿Qué es?	9
1.1.2. Open and Cross Innovation.....	10
1.1.3. ¿Quién hay?	10
1.2. Asterisk.....	11
1.2.1. Necesidades	11
1.2.2. Nuestra solución, Asterisk	12
1.2.3. Hardware	13
1.3. FreeBSD.....	19
1.3.1. Requisitos Asterisk	19
1.3.2. Que es FreeBSD	20
1.4. Análisis económico.....	22
1.4.1. Coste servidores.....	22
1.4.2. Coste teléfonos IP	22
1.4.3. Costes ATAs.....	23
1.4.4. Costes Gateway	24
1.4.5. Resumen de la inversión:	24
CAPITULO 2. IMPLEMENTACIÓN DEL MODELO.....	25
2.1. Instalación del servidor	25
2.1.1. Carga del sistema.....	25
2.1.2. Selección de idioma del sistema	25
2.1.3. Instalación de los datos del sistema.....	26
2.1.4. Configuración básica del sistema.....	27
2.2. Instalación de Asterisk	31
2.3. Instalación de Apache con módulos.....	31
2.4. Configuración del Servicio de telefonía.....	32
2.4.1. Funcionalidades	32
2.4.2. Usuarios.....	33
2.4.3. Codecs.....	33
2.4.4. Ficheros de configuración	34
2.5.5. Panel de operadora	43
2.6.. Integración en ESADECREAPOLIS.....	46
2.6.1. Descripción de la red.....	46
2.6.2. Integración de Asterisk	46
2.7.. Resultados de la telefonía	48

BIBLIOGRAFÍA 51

ANNEXOS..... 53

INTRODUCCIÓN

El origen de este TFC se sitúa en el 2007, enmarcado en un proyecto mucho más grande. Esade planificaba un macro proyecto que consistía en un campus universitario situado en Sant Cugat del Vallés con una facultad, una residencia universitaria, un parque de negocios y un Hotel. A día de hoy, en 2011, la facultad acoge diariamente a muchos alumnos de los que, al menos una porción, viven en la residencia universitaria mientras que en el mismo campus, un parque de la innovación cuenta con más de 70 empresas activas trabajando en su día a día.

Concretando, nació de uno de los muchos servicios IT que se planificaron para este parque de la innovación previamente comentado. La elaboración de un sistema de telefonía que diera servicio a todos los inquilinos y trabajadores.

Hoy en día, si nos ponemos a pensar, resulta imposible imaginar una empresa sin una infraestructura de comunicación de voz eficiente y estable. La imagen hoy día lo es todo, y buena parte de la imagen de las empresas se dan cada vez que realizan una llamada. Además, fomentar la comunicación interna y ofrecer un trato diferencial al cliente son premisas claves por tal de aumentar la competitividad de la empresa.

En pocas decenas de años hemos pasado de la conmutación de circuitos a la comunicación móvil de última generación, un cambio tan radical que es una barbaridad comparar ambas tecnologías

El ancho de banda creciente a nivel mundial y la optimización de los equipos de capa dos y tres para garantizar la calidad de servicio de, entre otros, los servicios de voz en tiempo real son ya unos de los muchos motivos que existen para apostar por esta tecnología. Si además tenemos en cuenta los costes que tiene una centralita IP delante de los que tiene una centralita convencional, la decisión está tomada por unanimidad, hoy en día el servicio de telefonía sólo tiene sentido proponerse basándolo en la voz por IP.

El desarrollo del proyecto tiene dos grandes bloques; el primero recoge la descripción de un servicio de telefonía IP y su viabilidad técnico económico, mientras que en el segundo se define con detalle la manera en que se ha manifestado el proyecto.

Para desarrollar el proyecto, el primer bloque tiene en cuenta la necesidad de implementar una solución de telefonía IP de alto rendimiento y bajo coste, una solución basada en el software Asterisk sobre un sistema FreeBSD. Este precisamente fue el motivo por el que escogí este TFC.

Pese a no conocer Asterisk, el reto personal consistía en preparar una solución basada en software libre que no restará ninguna funcionalidad respecto a sus rivales en el sector. Y poder ponerla a funcionar para que usuarios trabajaran con ella. Una motivación ciega y pragmática que me permitió formarme y conseguir un lugar de trabajo que a día de hoy conservo.

El lugar donde se ubicará el servicio de telefonía por IP es el parque de negocios ESADECREAPOLIS, se define como un centro de Open&Cross innovation promovido por la escuela de negocios ESADE, cuatro de las principales Cajas de Ahorros Catalanas, el Ayuntamiento de Sant Cugat y la Generalitat de Cataluña por medio de Avançsa. Acoge a muchas empresas y en sus instalaciones tienen lugar múltiples actividades enfocadas a facilitar procesos de innovación. El centro de innovación tiene como objetivo crear un ecosistema que contribuya a estimular la creatividad y la innovación en las empresas.

Por la definición del parque y la pequeña exposición de mis motivaciones, creo que es posible intuir los principales objetivos que tendrá el proyecto, o al menos los que marcábamos en el inicio del proyecto.

En primer lugar, la necesidad de implementar una solución de alto rendimiento esta impuesta. Hemos descrito un entorno con usuarios que provienen de diferentes empresas. Muchas personas que llegan teniendo experiencias con otras centralitas de voz por IP comunes en el entorno empresarial.

En segundo lugar, proponer una solución que no sea prohibitiva para ningún residente en el parque. La delicada situación que atraviesan, en general, las empresas actualmente implica que el servicio ofertado sea de bajo coste. El objetivo siempre tiene que ser ofrecer un valor añadido.

En tercer y último lugar, cumplir los dos anteriores y estar satisfecho del resultado de mi trabajo.

En lo que respecta al segundo bloque, se basará, en su mayoría, en descripciones técnicas sobre la tecnología que utilizaremos, en primera instancia explicare paso a paso como instalar un servidor FreeBSD y configurarlo. Una vez lo tengamos, instalaremos todo el software que sea necesario para el servicio y lo configuraremos con ejemplos fáciles de entender. Por último, haré un breve balance de cómo ha funcionado el servicio en líneas generales.

CAPÍTULO 1. MODELO INICIAL

1.1. Esadecreapolis

Si el objetivo de mi proyecto es crear un sistema de telefonía para un parque de negocios y sus inquilinos, es importante conocer el clima en el que convivirá esta integración para encontrar la solución que se adecue a sus clientes, en este punto intentaré explicar brevemente que es y quién hay en ESADECREAPOLIS, y la idea en la que intentaremos que nuestro servicio de telefonía cuaje en los usuarios.

1.1.1. ¿Qué es?

ESADECREAPOLIS es un parque de innovación donde empresas de múltiples sectores se alojan en un entorno de alto rendimiento, con el objetivo común de acelerar su innovación de una manera abierta y cruzada.

ESADECREAPOLIS está ubicado en Sant Cugat y cuenta con un edificio de última generación con 4 plantas y 240 metros de longitud, podemos verlo en la figura inferior.



Figura 1.1. Edificio ESADECREAPOLIS

1.1.2. Open and Cross Innovation

Tradicionalmente, las empresas llevan años generando ideas y conocimiento de manera autónoma, básicamente en función de sus capacidades/experiencias y desarrollando únicamente las ideas que llegan al mercado. Esta innovación, generalizada hasta hace pocos años se le conoce con el término “close innovation”. Podemos sintetizar la idea en la siguiente figura.

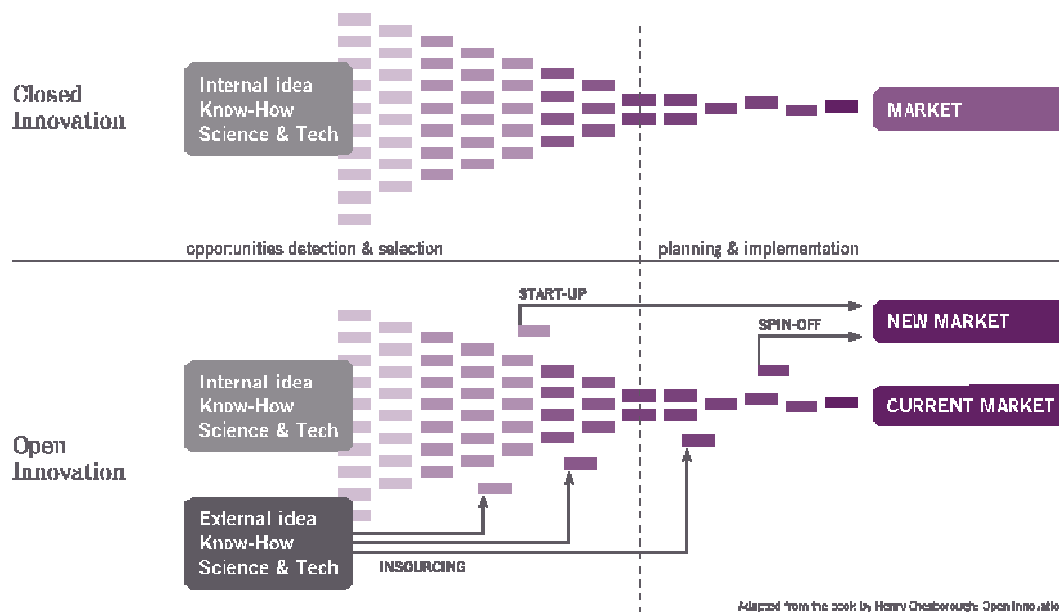


Figura 1.2. Comparativa de los métodos de innovación.

En el entorno multiempresarial de ESADECREAPOLIS, se abre y cruza la innovación entre sus compañías y otras, como partners o subministradores. Un ejercicio disruptivo que puede desembocar en nuevos conceptos de negocio más allá del mercado actual. Siempre con el objetivo de diversificar y aumentar el rendimiento empresarial.

1.1.3. ¿Quién hay?

Podemos encontrar desde empresas consolidadas como Roca, Laboratorios Esteve, Caixa Catalunya hasta emprendedores de renombre como Mykoots, Menta ITV, Flowlabs, Disfrutalia, Visual Century, Greenlabs, etc.

Sin embargo, creo que lo importante de este punto es como encajarán los servicios de telefonía con sus posibles usuarios. En este aspecto, diferenciamos dos tipos de clientes, el que se hospeda en el Business Center de ESADECREAPOLIS y el que se está alojado en un iLab:

El Business Center son espacios equipados para 2–6 personas, que se entregan llave en mano preparados para trabajar 24 horas. A los clientes de Business Center se le provee de un servicio completo de telefonía, de manera que cuando llegue, el cliente tendrá montado un teléfono listo para usarse.

En los iLabs se disfruta de un espacio diáfano, desde 60 a 1300 metros cuadrados. A estos clientes, se ofrecerá nuestra ayuda para que puedan integrarse con nuestra centralita.

Un punto fuerte de este proyecto es ayudar a crear un espacio más colaborativo y de rápido contacto, para fortalecer tecnológicamente la propuesta de ESADECREAPOLIS.

1.2. Asterisk

Llegamos al núcleo de la cuestión, la parte fundamental de los sistemas de telefonía es, sin duda, el software. El encargado de gestionar todo este software será la centralita telefónica.

En este apartado justificaré cuales son las premisas que nos han llevado a tomar la decisión de escoger Asterisk como centralita.

1.2.1. Necesidades

Dado que haré mi proyecto para ESADECREAPOLIS, su marco corporativo me obliga a basar el diseño en una apuesta innovadora, una apuesta que funcione y que no esté generalizada. Pese a que el departamento de IT no es el Business Core de la empresa, tiene como objetivo entregar un servicio diferencial a todas las personas que utilicen sus servicios; como son los propios trabajadores de ESADECREAPOLIS, los clientes VIP de nuestro Business Center, los residentes en nuestros iOffices, etc.

Hoy día, el sistema de telefonía es importante para la gran mayoría de PYMES e imprescindible para las grandes corporaciones, hasta el punto que perder una llamada puede suponer perder un cliente potencial. Con esta premisa, pensamos que ofrecer un servicio de telefonía a los huéspedes era una obligación que no podíamos ignorar. Ahora bien, ¿cómo tiene que ser este servicio? Un servicio que ofertamos en un entorno multisectorial, dirigido tanto a grandes empresas como a pequeños emprendedores, tanto a gente que llama al su otra sede en Barcelona como a su proveedor asiático, tanto a empresas con una capacidad de inversión a otras que su objetivo es bajar costes... y todo con una única solución centralizada, con la visión de la telefonía de bajo coste como una herramienta de alto nivel comunicativo.

Por todo esto, buscábamos una solución que cumpliera las siguientes condiciones: de fácil encaje, moderno, innovador, de alta gama, robusto, sencillo, competitivo, escalable y que sea fácilmente gestionable. La solución que pensamos pasa por una centralita IP basada en software libre llamada Asterisk.

1.2.2. Nuestra solución, Asterisk

El software Asterisk fue desarrollado por Marc Spencer, por entonces estudiante de Auburn (Alabama) que en 1999 creó la empresa “Linux Support Services”. A medida que esta empresa creció, se encontró con la necesidad de tener una centralita telefónica por tal de poder ofrecer servicio 24 horas. Ante la imposibilidad de adquirir una centralita convencional por sus elevados precios, Marc decidió crear una centralita IP basada en el lenguaje C# sobre un PC de Linux, Asterisk.

Actualmente Asterisk está acabando su versionado 1.8 (con el valor estimado de 8 millones de dólares americanos y más de 250.000 líneas de código), con una comunidad gigantesca (más de 800 desarrolladores trabajan para Asterisk al año) coordinada por el propio Marc y cuenta con la empresa Digium, que facilita tanto Appliances como el hardware que necesitamos para utilizar Asterisk. El paralelismo entre el nacimiento de Asterisk y nuestra idea es innegable, ojala el resultado a medio plazo también sea paralelo.

Para realizar el proyecto, utilizaremos la versión 1.6 de Asterisk, que incluye muchas características, las principales son:

- Buzón de voz con soporte de voicemail.
- Conferencias telefónicas.
- Posibilidad de configurar IVRs.
- Redirecciones.
- Grupos de llamadas.
- Captura de llamadas.
- Ejecución interactiva integrada en Dialplan.
- Paneles de operadoras.
- Soporte a protocolos VoIP SIP, H323, IAX y MGCP.
- Soporte DAHDI para hardware.
- Soporte para conexiones digitales LIBPRI.

La presentación al mercado de Asterisk se presenta en tres formatos:

- Software Asterisk abierto, libre y multiplataforma.
- Asterisk licensing es la modelización a medida de Asterisk para paquetes cerrados.
- Tribox, que lo entendemos con un Appliance que contiene un Asterisk listo para trabajar.

Para el proyecto, he escogido la primera opción por los siguientes motivos; es gratuita, me permite tener más flexibilidad para aprender cómo funciona Asterisk y puedo sacarle más rendimiento integrándolo en los sistemas que tenemos en producción en ESADECREAPOLIS, con lo que conseguiría prestaciones como automatizar parte de su mantenimiento.

De esta manera, Asterisk nos permitirá ofrecer multitud de servicios telefónicos que esperamos, sacie nuestras particularidades.

1.2.3. Hardware

Pese a que en un sistema de voz por IP el plato fuerte es el software, necesitaremos un hardware para poder entregar el servicio. En este punto intentaré definir las claves para escoger un buen hardware, son los pilares que mi servidor Asterisk utilizará para ofrecer esta herramienta comunicativa que buscamos.

Básicamente, necesitamos los siguientes elementos:

- Servidor: Necesitaremos una máquina que de soporte al servidor para poder instalar y poner en producción Asterisk. Dadas las características de Asterisk, sería ideal que se tratara de un servidor dedicado en lugar de un entorno virtualizado (disponibilidad total) y que estuviera redundado, por tal de tener un backup rápido en caso de avería o caída del otro servidor. Así pues, creo que la mejor opción es comprar dos equipos enracables que puedan estar alojados en un CPD, teniendo como opción económica la posibilidad de crear dos máquinas virtuales (una encendida y otra a la espera de una posible caída) en una granja de máquinas virtuales.
- Teléfonos IP: Por tal de explotar el servicio de voz sobre IP, necesitaremos terminales que soporten esta tecnología, los teléfonos IP. Los teléfonos IP son terminales que permiten realizar una comunicación utilizando una red IP, generalmente se trata de comunicaciones de paquetes de datos en los que se transporta voz o video. Estos teléfonos tienen un aspecto generalmente parecido a un terminal telefónico convencional, pero añaden un puerto donde conectamos el enlace con la red. También pueden tener algunas variaciones obvias como una pantalla más grande o a color.
- Atas: En los entornos empresariales nos topamos con servicios que actualmente están en proceso de desuso, sin embargo, todavía son comunes. Un ejemplo claro es el fax, que pese a estar en proceso de sustitución por el mail todavía es habitual en las empresas. Por tal de poder ofrecer servicio de fax necesitaremos un convertidor que nos permita pasar de la telefonía IP a la telefonía analógica. Esta solución pasa por un producto llamado adaptador para teléfono analógico (ATA, Analog Telephone Adapter). Aparentemente el ATA tiene el aspecto de una cajita con una entrada Ethernet i varios puertos de salida RJ11

telefónicos. Administrativamente, normalmente suelen gestionarse vía Web con alguna pequeña interfaz que te permite configurarlo.

- Teléfonos móviles: Por tal de dar un servicio más completo, uno de mis objetivos es que los clientes que posean terminales móviles de alta gama puedan llamar a través de mi centralita, como si se tratara de un teléfono interno más. Para ello, el móvil tiene que poder soportar el protocolo SIP, que más adelante detallaremos.

Enlace con la red telefónica conmutada: Para lanzar llamadas al exterior desde un sistema IP tenemos dos opciones, buscar un operador de llamadas por Internet o contratar algún acceso a la red telefónica conmutada. En ambas opciones necesitaremos un router que haga funciones de Gateway, en el primer caso para enrutar los paquetes a través de Internet y en el segundo caso para transformar la señalización SIP a una ISDN/PRI.

Después de definir el hierro que necesitaré para mi sistema de telefonía IP, ahora detallaré los equipos concretos que utilizaré para ofrecer este servicio.

1.2.3.1 Servidor

Como todo sistema escalable, las necesidades de nuestro servidor dependen principalmente del uso que se le tenga que dar. Si lo que queremos es un equipo de pruebas para llamar entre dos teléfonos IP cualquier equipo nos sirve, mientras que si queremos reproducir la telefonía IP que la marca McDonal's tiene implantada en California, con más de 1000 terminales, posiblemente necesitemos un equipo con más prestaciones.

Claramente, mientras más crece el sistema (más usuarios), más llamadas tendremos. Estas llamadas en aumento se reflejan en el servidor sobretodo en un aumento del uso de la CPU, pese a que también podemos observar que ocupa más memoria, el recurso de uso de CPU es el más crítico. Por lo que hemos consultado las recomendaciones de Digium en cuanto a uso de CPU.

Digium hace hincapié en este aspecto en su documentación, habla de que los puntos críticos de consumo de su sistema (transformación de CODECS, cancelaciones de eco, scripting concurrente del plan de marcado, etc.) atacan en gran medida a la CPU del sistema, concluyendo con la siguiente tabla que vemos a continuación.

Tabla x. Relación de necesidades de Asterisk.

Purpose	Number of Channels	Minimum recommended
Hobby system	No more than 5	400 MHz, 256 MB Ram
SOHO system	5 to 10	1GHz, 512 MB RAM
Small Business	Up to 15	3GHz, 1GB RAM
Médium to large system	More than 15	Dual CPU, 2GB RAM

Las expectativas del sistema son las de dar servicio a todas las personas que pueden llegar a hospedarse en ESADECREAPOLIS, las previsiones son 100 empresas de una media de 8 personas, por lo que contamos 800 personas, en todo caso estaríamos en el rango de sistema para mediana o gran empresa, por lo que el equipo que soporte el servidor tendría que tener una CPU doble con, al menos, 2GB de RAM.

Las opciones que tenemos sobre la mesa son las de un servidor de la gama PowerEdge de Dell contra la serie IBM Power 500, de prestaciones similares, mientras que son algo más económicos los equipos de DELL.

Teniendo en cuenta todo lo comentado, la elección final son dos servidores DELL PowerEdge R300, podemos verlo en la figura inferior. Contiene un procesador Quad Core Intel Xeon a 2.5GHz, 2x3M de cache, 4GB de memoria RAM, 160GB SATA a 7200 rpm por 836,60€ cada uno. Los dos equipos tendrán que estar sincronizados, de manera que si uno fallara, el servicio seguiría funcionando para el usuario.



Figura 1.3. Servidor DELL Power Edge

1.2.3.2 *Teléfonos IP*

En este punto, tengo que comentar que cualquier teléfono IP del mercado que soporte SIP es compatible con Asterisk y se podría integrar, con mejor o peor resultado, en nuestro sistema de telefonía sin necesidad de variar la configuración que en este proyecto presento.

Ahora bien, para mi proyecto he apostado por dos opciones, la diferencia entre ambas será la gama del producto. Cabe destacar que en ambos casos tengo que tener garantizado la robustez del teléfono, por este motivo he rastreado la gama de terminales Cisco y Polycom, en ambos casos son productos que tienen una arraigada tradición en el mundo de la telefonía IP y acumulan años ofreciendo buenos resultados.

Por otro lado, otro punto que juega a favor de esta elección es el dominio de Cisco en el mercado actual, creo que es muy buena idea tener la posibilidad de ofrecer a los clientes poder aprovechar sus teléfonos con nuestro sistema de telefonía.



Figura 1.4. Polycom SoundPoint IP 330

En conclusión, en mi proyecto utilizaré dos terminales, uno de coste reducido que podemos ver en la figura superior (Polycom SoundPoint IP 330, 120€) y otro de gama más alta (Cisco 7941, 250€) podemos verlo en la figura inferior, no hay que olvidar que la electrónica escogida soporta sin problemas toda la gama de teléfonos Cisco.



Figura 1.5. Cisco IP Phone 7941

1.2.3.3 Adaptador para teléfonos analógicos (ATA)

Este punto es similar al anterior, cualquier ATA que soporte SIP puede integrarse con Asterisk.

En cuanto a la selección, básicamente tengo tres premisas en mente; el precio, el número de líneas que necesitemos y que sea un producto que ofrezca garantías.

Si lo que buscamos es un ATA simple orientado a una o dos líneas, como podría ser el típico caso de la impresora multifunción que necesita un número externo, el producto ideal según mi opinión, que utilizaré en el proyecto son los Cisco ATA 186. Cisco ATA 186 provee dos líneas analógicas, con una entrada para Ethernet mediante la que podremos configurar el entorno de administración y pasar los datos, soporta SIP y los CODECS g729 y g711, comunes en la telefonía IP. Por otro lado es un producto muy robusto y su precio ronda los 80 €, podemos verlo en la imagen inferior.



Figura 1.6. Cisco ATA 186

En el otro extremo, si tenemos una necesidad masiva de líneas analógicas sobre nuestra centralita IP, tendremos que pensar en un equipo ATA más grande. Imaginemos el caso de una residencia de estudiantes, esta puede tener una solución IP basada en Asterisk con teléfonos IP para sus trabajadores. Sin embargo, el precio de proveer equipos IP para todos los estudiantes induce un alto coste con un gran riesgo, de esta manera tendremos que buscar un ATA que nos ofrezca líneas para los estudiantes. Para esta solución he escogido el producto Siemens Mediatrix 4124 que podemos ver en la figura inferior, este nos provee 24 puertos analógicos, soporta SIP y los CODECS usuales. El coste de este ronda los 1600 €.



Figura 1.7. Mediatrix 4124

1.2.3.4 *Teléfono móvil*

Para que un teléfono móvil pueda integrarse en nuestra centralita, tan sólo tiene que soportar SIP. Mi elección de teléfono es la del Nokia E71 dado que es el que usa todo el equipo de ESADECREAPOLIS.



Figura 1.8. Nokia E71

1.2.3.5 Gateway

Para enlazar con la red telefónica conmutada hemos contratado un acceso primario, que nos provee de 30 canales de entrada simultáneos para líneas fijas con Orange y un acceso de 15 canales para líneas móviles con Movistar.

El equipo que utilizamos para conectar estas líneas de primario es un router Cisco 2800 ejerciendo de Gateway con una tarjeta PRI E1, que podemos ver a continuación.



Figura 1.9. Cisco 2800



Figura 1.10. Tarjeta PRI-E1 para Cisco 2800

1.3. FreeBSD

Como ya hemos comentado anteriormente, una de las principales ventajas que nos ofrece la telefonía IP es la bajada de precios, sobretodo en llamadas a larga distancia (hasta 50 veces más económicas), que son usuales en las compañías. El avance del estudio de CODECS, que codifican la voz de una manera cada vez más compacta y el avance de las conexiones ADSL por parte de las operadoras en el mercado de las telecomunicaciones han permitido que los servicios PSTN-VoIP cada vez sean más frecuentes.

Además de los beneficios económicos que hemos comentado, VoIP nos permite “softwarizar” el servicio de telefonía por delante de los complejos montajes de las tradicionales centralita analógicas. Estos beneficios podrían convertirse en un arma de doble filo, si el software de nuestra centralita VoIP trabajara sobre un sistema operativo poco estable, por lo tanto la elección de nuestro SO (Sistema Operativo) es crítica.

1.3.1. Requisitos Asterisk

Para seleccionar el SO, he seguido las premisas de las especificaciones de Asterisk. Originalmente, Asterisk fue desarrollado sobre GNU/Linux, pero actualmente también se distribuye en versiones para los sistemas operativos FreeBSD, MacOSX, Solaris y Microsoft Windows. Aunque soporte todos estos, la propia documentación de Asterisk cataloga las versiones sobre BSD y Linux como “adecuadas” para sistemas de alta disponibilidad. Por ello, mis finalistas son FreeBSD y Linux.

La principal ventaja que presenta Linux no es otro que Asterisk fue desarrollada sobre él, sin embargo, Asterisk no deja de ser una aplicación basada en C++, donde FreeBSD es tan fiable como Linux.

FreeBSD tiene, en mi opinión, varias ventajas. Por un lado, FreeBSD tiene varias características que lo convierten en uno de los más populares a nivel de servidor y por usuarios profesionales, la buena reputación del sistema es tal

que entidades de primer nivel mundial como “Yahoo”, “DataPipe”, “Pair Networks” o “New York Internet” han reconocido públicamente utilizar FreeBSD. El propio sistema MacOSX está basado en FreeBSD porque según pública Apple en las Release Notes de sus eMacs “la disponibilidad que ofrece el sistema es única gracias a la madurez de su código base”. Estoy seguro que Linux tiene tantos avales como FreeBSD sino más, sin embargo, queda claro que FreeBSD es una válida.

Por otro lado, personalmente conozco mucho más FreeBSD que Linux, por lo que creo que, en igualdad de garantías, será mucho más fácil para mi desarrollar el proyecto sobre esta plataforma, además, profesionalmente la licencia BSD es mucho más atractiva para profesionales que la GNU, Apple puede dar fe de ello.

Por todo esto, pienso que FreeBSD es la plataforma adecuada para realizar el proyecto.

1.3.2. Que es FreeBSD

La distribución FreeBSD (Free Berkeley Software Distribution) tiene su origen en extensiones del sistema operativo UNIX de AT&T Research. Varios proyectos de código abierto salieron de la distribución conocida como 4.4BSD-Lite. FreeBSD añade un buen número de paquetes de otros proyectos de código abierto. Está representada por su mascota Bestie, que podemos ver en la figura de la derecha.

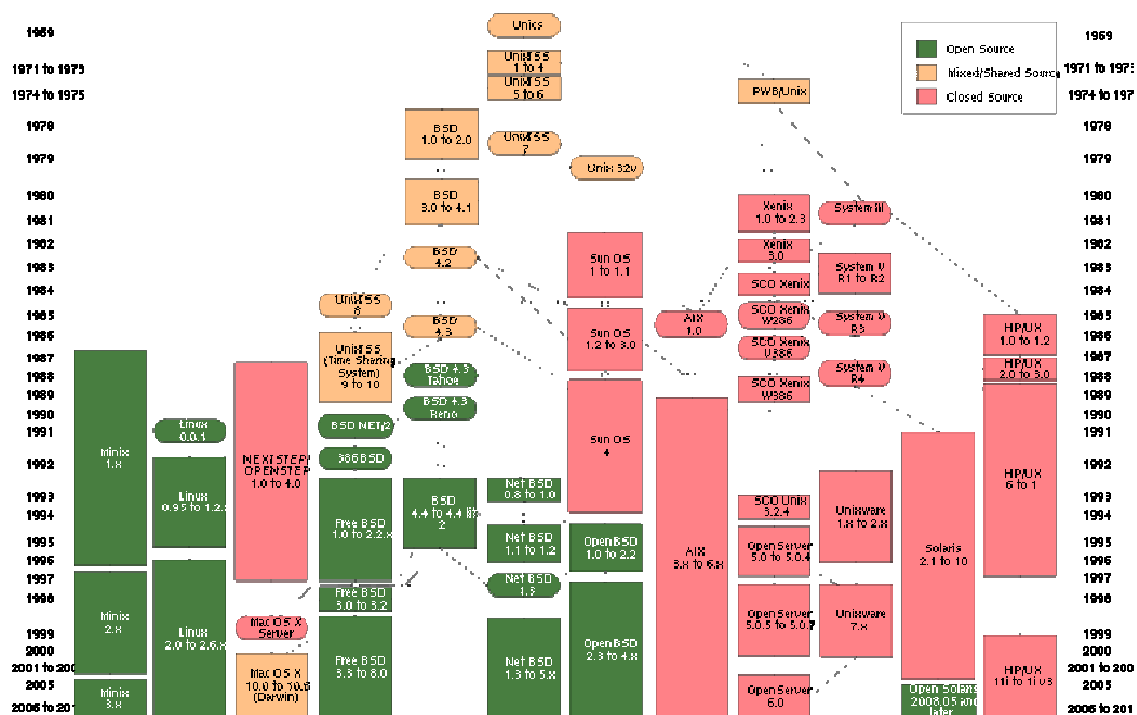


Figura 1.11. Historia de Sistemas Operativos provenientes de Unix

Básicamente, como podemos ver en la figura superior, FreeBSD está compuesto por el núcleo de 4.BSD-Lite, que se encarga de la planificación de ejecución de procesos (con capacidad multiproceso), la gestión de memoria, los controladores de dispositivos, etc.... Por otro lado, la API del sistema será la biblioteca C de Berkeley, incorpora la Shell de los proyectos GNU, el sistema X Window para el entorno gráfico (con soporte para KDE, Gnome y muchos más) y muchas otras aplicaciones.

El proyecto FreeBSD ha vivido un versionado continuo desde 1992, aunque la aparición de actualizaciones se ha incrementado desde 2003, como podemos ver en el siguiente gráfico.

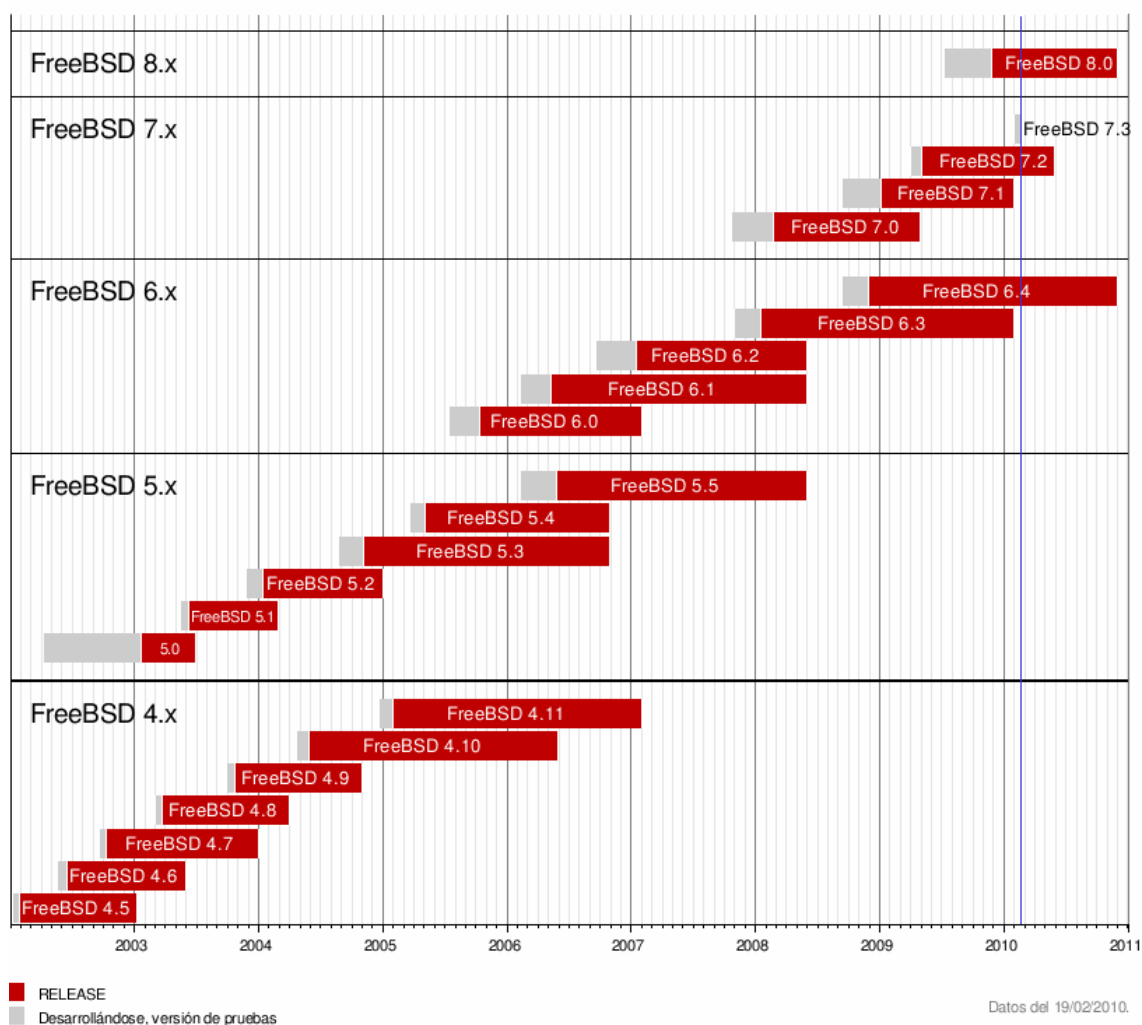


Figura 1.12. Planificación de versionados de FreeBSD.

En conclusión, para llevar a cabo el proyecto, utilizaré la colección de puertos de FreeBSD, que proporciona de una manera intuitiva y consistente de instalar aplicaciones en FreeBSD. Utiliza ficheros Makefile para descargar el código de las aplicaciones desde la red o el disco local, compilar e instalar estos nuevos paquetes con un criterio continuista y gestionando todas las dependencias del paquete en cuestión.

1.4. Análisis económico

El impacto económico del proyecto era la pieza básica para que pudiéramos realizarlo, teníamos que conseguir que rebajará a su símil con otras plataformas de más fácil acceso y con amplia tradición (Cisco, Avaya,...). Por todo esto intentamos optimizar los costes en todos los apartados del presupuesto.

El esbozo que queremos llevar a cabo se compone de diferentes apartados. En primer lugar tenemos el coste de los servidores que nos tienen que ofrecer el servicio, en segundo el coste de los teléfonos que ofreceremos a los clientes, en tercer lugar el precio de los transformadores digital-analógico de líneas y, por último, el coste de los primarios que nos permitirán conectar con la red telefónica conmutada. Todo esto considerando que el precio de la infraestructura que proveerá de comunicaciones el edificio esta asumida en otra partida, por lo que no la tendremos en cuenta en esta previsión.

1.4.1. Coste servidores

En primer lugar los servidores. La centralita es la pieza básica del sistema, todo será gestionado a través de ella por lo que es de vital importancia. Gracias a la optimización de Asterisk sobre FreeBSD, demostrada en las pruebas que realizamos previamente, consideramos que no necesitaremos unos requisitos muy extensivos y buscando el equilibrio apostamos por los dos servidores DELL de gama media comentados en el apartado 3.3, que nos costarán un total de 1000 € cada uno.

1.4.2. Coste teléfonos IP

El teléfono es la parte del sistema que interactuará con nuestros clientes, por lo que decidimos apostar por los Cisco Iphone 7941, descritos en el anterior apartado, dado que eran los que más garantías nos ofrecían. Estos cuestan 180€ cada uno.

Dado que el total de los teléfonos será una de las partidas importantes de la estimación, ayudaría mucho que dividiéramos en dos la inversión de estos. Para ello, se nos ocurrió que no hacía falta comprar teléfonos para llenar todos los despachos si no estaba prevista una ocupación completa en el arranque, por este motivo dividiremos la inversión en dos partes. Una se ejecutará al arranque del edificio y otra al final del primer año.

La primera recoge los teléfonos del personal de ESADECREAPOLIS. Se trata de 22 teléfonos distribuidos de la siguiente manera:

- 1 para presidencia
- 1 para dirección
- 4 para el dep. de operaciones
- 4 para el dep. de innovación
- 2 para el dep. comercial
- 2 para el dep. de marketing
- 4 para el dep. de mantenimiento
- 4 para el dep. de informática

Por otro lado, el Bussines Center tiene cinco salas de reuniones alquilables para los clientes, que tenemos cubrir con teléfonos para hacerlas más confortables para las personas que las utilizan. Estos cinco teléfonos también serán cubiertos con la primera inversión.

El edificio tiene un parking que es gestionado desde una habitación, en esta cabina también proveímos un teléfono para que los trabajadores que controlen los accesos puedan comunicarse.

Por último los teléfonos dedicados a los clientes directamente. Para prever los teléfonos que necesitaré he tenido en cuenta los dos tipos de despachos que tenemos en el Bussines Center. Para los despachos pequeños necesitaremos dos teléfonos mientras que para los grandes necesitaremos un total de cuatro.

Al arranque del parque teníamos contratados tres despachos de los pequeños y cinco de los grandes. Esto significa veintiséis teléfonos de entrada, además dejaremos una previsión de doce teléfonos más para las llegadas de nuevos inquilinos el primer año en el Bussines Center o en algún iSuite.

Con todo ello necesitaremos para arrancar un total de sesenta seis teléfonos, lo que supondrá la inversión de once mil ochocientos ochenta euros (11880€).

Para la segunda inversión quedarán un total de setenta teléfonos para comprar, lo que significara doce mil seiscientos euros (12600€).

1.4.3. Costes ATAs

En cuanto a líneas analógicas, he contado que sólo necesitaremos tener la línea del fax de la empresa, esto equivale a un convertidor Cisco ATA que tiene un coste de 90 €.

Además, esta opción me ha parecido muy interesante para ofrecer a empresas, tiene un coste reducido, por lo que podría ser interesante incorporarlo como servicio opcional. Un servicio de fax a bajo coste con mantenimiento inferior a una línea analógica.

Además, el edificio tiene ascensores que tienen que estar dotados de líneas de teléfono, estos tenía previstos cubrirlos con ATA's, sin embargo, la normativa obliga que estas líneas tienen que ser directas a un operador, por lo que no hemos podido aprovechar ATAs para cubrir esta necesidad.

1.4.4. Costes Gateway

Para cubrir la conexión con la red telefónica conmutada propusimos comprar un Gateway Cisco con posibilidad de incorporar tarjetas de primarios de operadoras. El coste del Gateway es de 4000€, más una previsión de mantenimiento de un 10% de su valor nominal. En cuanto a las tarjetas para primarios hemos llegado a un acuerdo con Movistar y Orange para que nos las proporcionen a coste 0, teniendo 30 canales simultáneos por cada primario.

1.4.5. Resumen de la inversión:

Tabla 1.13. Estimación de costes para el proyecto.

Concepto	Inversión inicial arrancada	Inversión a un año vista
2 servidores	2000€	
Teléfonos IP	11880 €	12600€
ATAs	90 €	
Gateway Cisco	4000€	
Enlace operador IP	0€	
Primario Orange	0€	
Primario Movistar	0€	
TOTAL	17970€	12600€

El total de la inversión que hay calculado es de diecisiete mil novecientos setenta euros el primer año y de doce mil seiscientos el segundo, entro los dos un total de treinta mil quinientos setenta euros (30570€).

CAPITULO 2. IMPLEMENTACIÓN DEL MODELO

2.1 Instalación del servidor


En este punto intentaré simplificar el proceso de instalación del servidor que necesitaremos y todos los servicios que alojaremos en él. Para complacer los objetivos que nos hemos marcado en nuestro servicio de telefonía necesitaremos los siguientes elementos:

- FreeBSD 8 y accesibilidad (ssh y ftp).
- Asterisk.
- Servidor Web (Apache22), php y perl.
- Servidor TFTP.

Para instalar FreeBSD, accederemos a su Web y descargaremos la versión 8, que es la última estable que presenta FreeBSD y tiene soporte completo para Asterisk.

2.1.1 Carga del sistema

La descarga en formato iso, y se debe grabar en un cd para arrancar desde este medio. El sólo cargara un asistente de instalación del sistema operativo como cualquier Linux o Windows XP. En la captura de debajo tenemos un ejemplo que como FreeBSD reconoce y carga los dispositivos para poder instalarse en el equipo.



```
CD Loader 1.2
Building the boot loader arguments
Looking up /BOOT/LOADER... Found
Relocating the loader and the BTX
Starting the BTX loader

BTX loader 1.00  BTX version is 1.02
Consoles: internal video/keyboard
BIOS CD is cd0
BIOS drive A: is disk0
BIOS drive C: is disk1
BIOS 639kB/129984kB available memory
```

Figura 2.1. Carga FreeBSD

2.1.2 Selección de idioma del sistema

Como podemos ver en la captura inferior, lo primero que preguntará el asistente de instalación es en qué región se encuentra el equipo que se va a

instalar, esta opción es importante por tal de adecuar el teclado y la franja horaria.



Figura 2.2. Instalación FreeBSD

2.1.3 Instalación de los datos del sistema

En este punto llega el momento de instalar los archivos que compondrán propiamente el sistema. Un sistema que ofrezca servicios como servidor no tiene que estar sobrecargado con elementos que no necesite, tenemos que aumentar lo máximo que podamos su robustez y ligereza.

Por este motivo, es importante hacer una configuración personalizada para el sistema operativo. Por lo que tendremos que seleccionar la opción "Custom" del asistente tal como vemos en la figura superior, apareciendo un menú que podemos ver en la imagen inferior.

En este podemos configurar las opciones varias del sistema (DHCP, Debug del sistema, editor de texto por defecto, etc.) de instalación, Podemos definir particiones y sus etiquetas, escoger entre unas instalaciones de paquetes predefinidas y el medio mediante el cual se instalaran los paquetes.

En mi caso, he decidido separar la en dos particiones diferentes la memoria Swap del servidor y la memoria de datos. He seleccionado Kern Developer como precarga de paquetes, dado que creo que es la que más se adapta a nuestras necesidades. Lo podemos ver en la captura inferior.

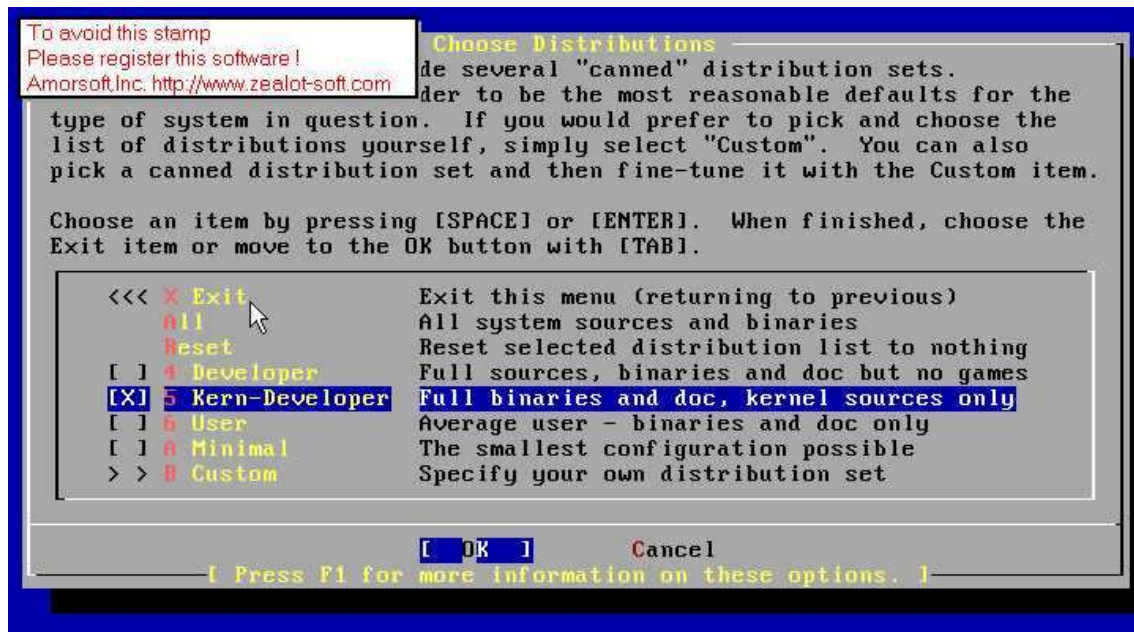


Figura 2.3.. Instalación FreeBSD.

Por último, el medio que he escogido para la instalación es el del cd, se procederá a la instalación del sistema copiando los archivos que necesite del cd. Cuando esta acabe, al rebotar el sistema desde la partición de datos que hemos configurado anteriormente ya podremos cargar el sistema operativo y proceder a configurarlo adecuadamente.

2.1.4 Configuración básica del sistema

Ahora detallaré una configuración básica de FreeBSD para que el servidor pueda acoger el sistema de telefonía Asterisk. Me centraré en los siguientes servicios:

- Configuración básica de red.
- Asignar DNS
- Definir el host
- Servicios ssh, ftp y tftp.
- Copia de archivos críticos del sistema.
- Actualización de ports.
- Servidor DHCP

Para configurar la red, tendremos que hacer servir los siguientes comandos.

```
ifconfig eth0 inet direccion_IP netmask mascara_de_red
route add default router_por_defecto
```

Figura 2.4.. Configuración de interfaz de red en Unix

En este caso;

```
ifconfig eth0 inet 192.168.6.12 netmask 255.255.255.0  
route add default 192.168.6.100
```

Figura 2.5.. Ejemplo configuración interfaz de red

Además, por tal que cuando se reinicié el servidor no perdamos la configuración tendremos que añadirla en el script de inicio de sistema que se encuentra en `/etc/rc.conf`

Por tal de definir los DNS que tiene que usar el servidor para poder resolver direcciones externas, imprescindible para actualizar el sistema o instalar Asterisk, tenemos que editar el archivo `/etc/resolv.conf` con el siguiente contenido.

```
domain Nombre_dominio  
nameserver IP_servidor_DNS_1  
nameserver IP_servidor_DNS_2
```

Figura 2.6. Configuración DNS

En este caso;

```
domain esadecreapolis.com  
nameserver 192.168.7.13  
nameserver 192.168.6.14
```

Figura 2.7. Ejemplo configuración DNS

Ahora tenemos que darle nombre al sistema, para ello editamos el archivo `/etc/hosts` con el siguiente contenido.

```
127.0.0.1 localhost localhost.dominio  
ip_servidor nombre_equipo nombre_equipo.dominio  
ip_servidor2 nombre_equipo2 nombre_equipo2.dominio
```

Figura 2.8. Configuración de hosts

En este caso;

```
127.0.0.1 localhost localhost.esacreapolis.com  
192.168.6.12 asterisk asterisk.esadecreapolis.com
```

Figura 2.9. Ejemplo de hosts

Por tal de poder enviar y recibir información necesitamos un usuario diferente de root en el sistema, que podemos crear a partir del comando `adduser` del sistema. Por otro lado, para los servicios de ssh, ftp y tftp utilizaremos el súper servidor `inetd`, que está integrado y preconfigurado en FreeBSD, para ello

añadiremos la comanda `inetd_enable="YES"` en `/etc/rc.conf` y, posteriormente, editaremos el fichero `/etc/inetd.conf` con la siguiente información:

ftp	stream	tcp	nowait	root	/usr/libexec/ftpd	ftpd -l
ssh	stream	tcp	nowait	root	/usr/sbin/sshd	sshd -i -4
tftp	stream	tcp	nowait	root	/usr/sbin/tftp	tftp -d dir

Figura 2.10.. Configuración `inetd.conf`

Para lanzar el servicio tan solo tenemos que lanzar la comanda `inetd`. Sin embargo el servicio `ssh` necesita que se reinicié la máquina, para poder negociar las claves pre-compartidas.

Cabe comentar que al arrancar `tftp`, el directorio que le marquemos será muy importante para los teléfonos Cisco. Estos utilizarán este servicio para cambiar de firmware y para coger la configuración que le asignemos al teléfono según su mac.

En el caso del cambio de firmware del teléfono, sólo tendremos que tener los archivos pertinentes en la carpeta de `tftp` y apretar una combinación de teclas en el teléfono para que este cambie el firmware por el que le pongamos.

Para el segundo caso, cada vez que arrancan los teléfonos van a buscar un fichero que se llama SEP seguido de la Mac del teléfono, dado su tamaño, he colocado este archivo en el anexo, aunque seguidamente añadiré una pincelada de la parte más importante. En caso que el teléfono no encuentre este archivo, se cargará con la última configuración válida.

```
<featureID>9</featureID>
  <featureLabel>5705</featureLabel>
  <proxy>192.168.6.12</proxy>
  <port>5060</port>
  <name>5705</name>
  <displayName>5705</displayName>
  <autoAnswer>
    <autoAnswerEnabled>2</autoAnswerEnabled>
  </autoAnswer>
  <callWaiting>3</callWaiting>
  <authName>5705</authName>
  <authPassword>1234</authPassword>
  <sharedLine>false</sharedLine>
  <messageWaitingLampPolicy>1</messageWaitingLampPolicy>
  <messagesNumber>5999</messagesNumber>
  <ringSettingIdle>4</ringSettingIdle>
  <ringSettingActive>5</ringSettingActive>
  <contact>5705</contact>
  <forwardCallInfoDisplay>
    <callerName>true</callerName>
    <callerNumber>true</callerNumber>
    <redirectedNumber>false</redirectedNumber>
```

```
<dialNumber>true</dialNumber>
</forwardCallInfoDisplay>
```

Figura 2.11. Fragmento de archivo SEP de un teléfono Cisco

Una vez acabado el inciso, existe un comando llamado portsnap que nos ofrece el servicio de actualizar los ports de FreeBSD, en la primera instalación del sistema es imprescindible actualizarlos, dado que de lo contrario podríamos tener problemas de integración de las aplicaciones que instalemos. Básicamente, “portsnap” tiene tres comandas; fetch, extract y update. “Portsnap fetch” descargará los últimos paquetes de ports que posee el FreeBSD en sus repositorios oficiales. Después “portsnap extract” los extraerá en las carpetas indicadas dentro de tu sistema. Y por último, “portsnap update” sobrescribirá los ports nuevos.

Por tal de instalar un servidor DHCP en nuestro sistema, tenemos que ir al port indicado, que se encuentra en el directorio /usr/ports/net/isc-dhcp3-server. Una vez dentro tenemos que tirar la comanda “make install”. El sistema, autónomamente, instalará el servicio. Podremos ver que se descargan los archivos de los repositorios en un directorio llamado distfiles, donde de querer reinstalar la aplicación podrá hacerse offline.

Para habilitar el servicio de DHCP tenemos que añadir la línea “dhcpd_enable=“YES”” en el archivo /etc/rc.conf. Para configurarlo, tenemos que crear un archivo en “/usr/local/etc” llamado dhcpd.conf que contenga la siguiente información.

```
option domain-name “nombre dominio”
option domain-name-servers ip_servidor_de_dominios
default-lease-time 600;
max-lease-time 7200;
ddns-update-style none;
subnet red_ip_asignables netmask mascara_grupo{
    range rango_ips_asignables;
    option ntp-servers ip_servidor_ntp;
    option tftp-server-name “ip_servidor_tftp”;
    option routers ip_router;
}
```

Figura 2.12. Configuración DHCP

En mi caso, haré servir el router que tenemos en la red para configurar el servicio DHCP, dado que no quiero añadir más DNS a nuestra red.

2.2 Instalación de Asterisk

Para instalar Asterisk utilizaremos los ports actualizados, nos colocamos en el directorio “/usr/ports/net/asterisk” y realizamos un “make install”.

En la instalación de Asterisk no se incluyen los addons de Asterisk, si queremos instalarlos solo tendremos que repetir la operación con el port de los addons de Asterisk.

Durante la instalación, se presentará un menú donde podremos definir que paquetes y dependencias queremos instalar, lo podemos ver en la imagen inferior. Como vemos, tenemos que seleccionar todos los paquetes dado que los necesitaremos con el tiempo.

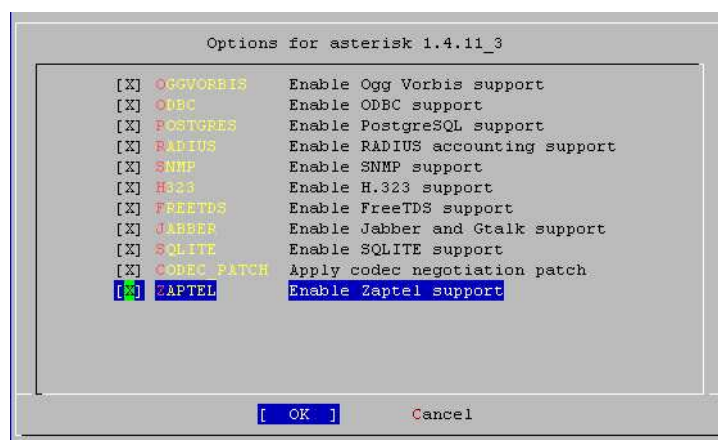


Figura 2.13.. Instalación Asterisk.

Además, para que cuando arranque el sistema arranque Asterisk es imprescindible añadir en el archivo “/etc/rc.conf” las siguientes líneas.

```
zaptel_enable="YES"
asterisk_enable="YES"
```

2.3 Instalación de Apache con módulos.

Para la operadora de Asterisk necesitaremos una plataforma de servidor Web, dado que estamos trabajando con FreeBSD, utilizaremos un apache y sus módulos para leer php.

Para instalarlo, tan sólo tenemos que dirigirnos al directorio /usr/ports/www/apache22 e realizar la comanda “make install”.

Apache 2,2 incorpora de serie el módulo SSL, por lo tanto tan sólo tendremos que instalar el módulo de perl y el de php.

Cuando esté instalado apache, tendremos que añadir al fichero rc.conf la línea `apache22_enable="YES"` para que, en caso del reinicio de la máquina, se encienda de nuevo.

Para instalar el módulo de Perl, nos dirigiremos a `/usr/ports/www/mod_perl2` y ejecutaremos `make install`.

Para instalar el módulo de php nos dirigiremos a `/usr/ports/lang/php5` y realizaremos el comando `make config`, cuando nos aparezcan todas las opciones tendremos que marcar con un tick la opción `mod_php5`. Después sólo quedará realizar el comando `make install`.

Para finalizar la configuración, tendremos que entrar en `/usr/local/etc/apache/httpd.conf` y añadir las siguientes líneas:

```
Loadmodule php5_module libexec/apache/libphp5.so
AddModule php5_module.c
<IfModule mod_php5.c>
    DirectoryIndex index.php index.html
</IfModule>
<IfModule mod_php5.c>
    AddType application/x-httpd-php .php
    AddType application/x-httpd-php-source .phps
</IfModule>
```

Figura 2.14. Ejemplo configuración módulo php

2.4 Configuración del Servicio de telefonía

En este punto explicare las nociones básicas para configurar una centralita en Asterisk y las específicas para cada alta en nuestro sistema. Para ello, primero describiré las funcionalidades del sistema de telefonía, después comentaré nociones muy básicas de los archivos de configuración que tiene Asterisk y acabaré con la descripción de la centralita que montaremos para ESADECREAPOLIS.

2.4.1 Funcionalidades

Tal como hemos comentado anteriormente, con la configuración que describo en este apartado todos los usuarios tendrán un servicio de telefonía con muchas funcionalidades, que comento seguidamente:

- Funciones básicas de llamada: Realizar y responder a una llamada, dejarla en silencio, marcar el último número llamado
- Llamada en espera: Podemos dejar en espera con música una llamada.
- Transferencia de llamadas: Podemos pasar las llamadas a cualquier otro teléfono.
- Multiconferencia: Podemos hablar con más de una persona en la misma llamada.

- Desvío de llamadas: Podemos hacer que cualquier llamada que reciba un teléfono se derive a cualquier otro.
- Captura de llamadas (Pickup): Si, estando en la misma oficina, oímos que llaman a un compañero que no está, podemos capturar la llamada desde nuestro teléfono.
- Voicemail y buzón de voz: Podemos escuchar los mensajes a través del teléfono o un email que nos enviará el sistema.
- Colas y agentes: Podemos definir estructuras con varios teléfonos recibiendo llamadas con diferentes estructuras lógicas.
- Salas de conferencia: La sala de conferencia es un ente que, todo el mundo que le llame se le preguntará un código, si lo sabe podrá entrar en un canal en el que podrá conversar con todo su equipo de trabajo.
- IVR (Interactive Voice Response): Los usuarios tendrán la posibilidad de configurar una IVR, que es un sistema telefónico capaz de recibir una llamada e interactuar con humanos mediante grabaciones de voz.
- Panel de Operadora: El panel consistirá en una Web que mostrará el estado de los teléfonos de la empresa.

2.4.2 Usuarios

Para realizar la adaptación a los requisitos de la empresa, plantearemos un plan de marcado que englobe a todos los clientes y la descripción de que funciones tendrán. Para describir los tipos de usuarios Asterisk posee tres formatos predefinidos; Los usuarios, los peers y los amigos:

- Usuarios: Cuando definimos un usuario, nos referimos a cualquier ente al que permitimos conectarse al servidor. Esto engloba a usuarios, teléfonos, ATAs, etc. Esto se utiliza para crear un canal que nos permita recibir llamadas y requiere una definición de contexto para indicar por donde entrarán las llamadas en nuestro plan de marcado
- Peers: Los peers no son más que una definición de conexión. Es decir, cada peer tendrá que tener asociada una IP y se tendrá que autenticar en el servidor Asterisk para poder realizar/recibir flujos. En términos generales, se podría relacionar con los usuarios de la siguiente manera; los usuarios llaman al servidor mientras que es el servidor el que llama a los peers.
- Amigos: Una entidad que posea privilegios de Usuario y peer a la vez se definirá como Amigo.
Siguiendo con la definición, este podrá realizar y recibir llamadas.

2.4.3 Codecs

La conversión de ondas analógicas a información digital se realiza mediante codificadores/decodificadores (CODECS).

Asterisk necesita conocer diferentes tipos de codificadores para poder servir el audio de la conversación. Los CODECS soportados por Asterisk se clasifican, en su mayor parte según tres parámetros básicos:

- Bitrate: Cantidad de bits de información que se envía
- Sampling rate: Frecuencia de muestreo del señal
- Frame Size: Cada cuantos milisegundos enviamos paquetes con toda la información sonora.

Los CODECS más utilizados por Asterisk son los siguientes:

- GSM: Estándar de la ETSI, funciona mediante KPE_LTP (Regular pulse excitation long term predictor) con un bitrate de 13Kbps. Utilizado frecuentemente en las llamadas a móviles.
- G.729: Estándar de la ITU-T, trabaja a 8Kbps utilizando CS-ACELP(conjugate-structure algebraic-code-excited lineal-prediction). Tiene una variación llamada 729a que ofrece menos calidad con un bitrate inferior.
- G711: El CODEC de la RTB. Estandarizado por la ITU-T. Funciona con PCM y un bitrate de 64Kbps. Se presenta en Asterisk con dos versiones, ulaw y alaw.

Por defecto, Asterisk trata GSM y G711, sin embargo no g729. Este CODEC es muy competitivo por su moderada calidad frente su bajo bitrate. Dado que todos los dispositivos que utilizaremos lo pueden hacer servir teníamos que conseguir que nuestro servidor fuera también capaz de trabajar con él.

El primer escollo es que g711 es un privado y sujeto a licencia, después de verificarlo vimos que no era un problema dado que todos los teléfonos y el Gateway incluyen la licencia con su compra, por lo que no tendremos que pagar nada para añadir la funcionalidad.

Técnicamente, para que Asterisk pueda funcionar con g729 tan sólo debemos copiar la librería g729-gcc-pentium3.so (nos lo descargamos de la Web del estándar) en el directorio /usr/local/lib/asterisk/modules y reiniciar el servidor.

2.4.4 Ficheros de configuración

Asterisk tiene muchos ficheros de configuración, para no extenderme explicaré solamente los que se necesitará para definir el servicio de telefonía que tenemos como objetivo implementar.

Los ficheros que explicaré brevemente son los siguientes:

- sip.conf
- voicemail.conf
- extensions.conf
- agents.conf

- queues.conf
- meetme.conf
- manager.conf
- cdr_custom.conf

Empezaremos con el primero de ellos.

2.4.4.1 sip.conf

Tal como menciona su nombre, en este archivo definiremos todos lo que tenga que ver con el protocolo SIP.

La definición básica de un elemento en sip.conf tiene esta apariencia:

```
[general]
context=default
port=5060           ;//Puerto UDP.
bindaddr=0.0.0.0    ;//Escuchar por todas las direcciones IP del servidor
svrlookup=yes       ;//habilitar el servidor DNS SRV

[5077]              ;// Nombre de usuario
type=friend         ;// tipo de usuario
secret=jord1        ;//Pass para autenticar al amigo
callerid=Jordi<5077> ;//Nombre usuario y extensión
qualify=yes         ;//Tiempo de latencia no superior a 2000ms.
nat=yes             ;//El teléfono utiliza nat por el servidor.
host=dynamic        ;//IP dinámica
careinvite=yes      ;//Se permiten conexiones RDP
context=internal    ;//Es el contexto por el cual entrarán las llamadas en
el dialplan
group=1             ;//Definición del grupo
pickupgroup=1,2     ;//Definición de los grupos de recojida de llamadas.
```

Figura 2.15.. Ejemplo sip.conf

El fichero sip.conf empieza con la sección [general], que contiene la configuración por defecto de todos los usuarios y peers. Se pueden sobrescribir las variables por defecto en las configuraciones de cada usuario o peer.

Cada extensión está definida por un usuario, un peer o un amigo y con un nombre entre claudators.

Se puede monitorizar la latencia entre Asterisk y el teléfono/punto en cuestión mediante el parámetro “*qualify=YES*” para determinar si el dispositivo se puede utilizar, para este caso, Asterisk determina que para que un dispositivo sea operativo tiene que tener una latencia inferior a 2000ms. Si una extensión se encuentra detrás de un NAT, le podemos pedir a Asterisk que lo tenga en cuenta con el parámetro “*NAT=YES*” (ignora campos como la información de contacto e utiliza la dirección de la que proceden los paquetes).

El campo “*host=dynamic*” es muy importante en términos de seguridad. Significa que el teléfono puede conectarse desde cualquier IP, esto se puede limitar mediante una IP fija o un nombre de dominio.

El parámetro “*careinvite*” se refiere a que cuando dos usuarios han establecido conexión entre ellos, los paquetes RTP de audio se envían directamente entre ellos sin pasar por el servidor Asterisk. No hace falta comentar que esta propiedad entra en conflicto con otras como el NAT. En caso que los dos estén activos, Asterisk dará prioridad a *careinvite* para garantizar la comunicación.

El parámetro de “*context=default*” marca donde entrara la llamada que provenga de este elemento sip en el plan de marcada, se detallará más adelante en el fichero de configuración *extensions.conf*.

2.4.4.2 *voicemail.conf*

El fichero de configuración *voicemail.conf* sirve para configurar el contestador automático y gestionar los buzones de los usuarios.

También controla parámetros básicos que se aplican cuando tienen que grabar los mensajes de voz del contestador, como pueden ser el CODEC con el que se graba o los tiempos mínimos/máximos que puede durar el mensaje.

Hay dos contextos especiales, “*general*” y “*zone messages*” siempre tienen que estar añadidos.

```
[general]
attach=yes;           //Envia notificaciones por email
format=wav;           //Utiliza formato .wav para mensajes de voz
maxmessage=180;       //Limita el tiempo máximo a 180 segundos
minmessage=3;         //Limita el tiempo mínimo de mensaje a 3 segundos.
saycid=yes;           //Anuncia el número que ha llamado antes de que salte el
mensaje
maxlogins=3;          //Limita el máximo de intentos de log a 3
mailcmd=...;          //Fija la ruta del servidor de correo que utilizará Asterisk
servermail=...;       //Indica el origen del que vendrán los correos.
maxmsg=20;            //Limita el número de mensajes por buzón de usuario
operator=yes;         //Permite marcar una extensión cuando ha saltado el
contestador
fromstring=...;       //Modifica el “from” del mensaje de aviso de correo.
emailsubject=...;     //Modifica el “asunto” del mensaje de aviso
emailbody=...;        //Modifica el “contenido” del mensaje de aviso
```

Figura 2.16. Contextos fijos en *voicemail.conf*

Este último define zonas horarias. La hora para diferentes usuarios no es la misma y se tiene que poder informar sobre la hora de recepción de las llamadas. El formato con el que definimos las zonas es el siguiente:

zona=Pais/Ciutat|opcions

```
[zona messages]
madrid=Europe/Paris|'vn-received'Q'digits/at'R
```

Figura 2.17. Definición zona horaria

Cabe destacar que no tenemos que tocar estos campos para que funciones Asterisk, si no los tocamos, sencillamente funcionará con los parámetros por defecto.

Después de los dos contextos anteriores. También definiremos los buzones de los usuarios, se pueden tener todos en un mismo contexto o bien en diferentes.

El formato será el siguiente:

```
[default]
extensión=>password,nombre de usuarioi, e-mail,opciones
```

Figura 2.19. Formato de definición de voicemails

2.4.4.3 *extensions.conf*

Este fichero de configuración es el más importante de Asterisk, en él reflejaremos el plan de numeración de la centralita telefónica para cada contexto y usuario.

Como todos los archivos de configuración de Asterisk, extensions se divide en contextos que se marcan con su nombre entre corchetes y finalizarán cuando empiece otro contexto. Como en todos, cada uno de ellos tendrá su configuración independiente.

Hay dos contextos especiales reservados por el sistema que siempre, se utilicen o no, tienen que estar establecidos. Estos son el [general] y el [globals]. En el primero se configuran las opciones que tendrán todos los contextos y en [globals] tendremos todas las opciones que podrán ser utilizadas en el resto de contextos.

En el resto de contextos, todas las líneas tienen el mismo formato:

```
exten=>extensión,prioridad,comando(parámetros)

exten hace referencia al número marcado.
La prioridad al orden en el que se ejecutan los comandos.
El comando es la acción que se realizará.
```

Figura 2.20. Ejemplo configuración extensions

Ahora que ya conocemos el formato de las entradas en el dialplan, como hay demasiados comandos para mencionarlos todos, pondré un seguido de ejemplos que considero clave para entender como funciona extensions.

En el siguiente cuadro podemos ver la configuración que aplico para tratar las llamadas internas en ESADECREAPOLIS:

```
exten => s,1,Dial(${ARG1},20)
exten => s,2,Goto(s-${DIALSTATUS},1)
exten => s-NOANSWER,1,VoiceMail(su${MACRO_EXTEN}@voicemail)
exten => s-NOANSWER,2,Hangup()
exten => s-BUSY,1,VoiceMail(sb${MACRO_EXTEN}@voicemail)
exten => s-BUSY,2,Hangup()
exten => _s.,1,Goto(s-NOANSWER,1)
```

Figura 2.21. Ejemplo extensions.conf

En la primera línea, podemos ver que las llamadas se marcan con la misma extensión a la que vienen dirigidas, con esto, haremos sonar el teléfono de la persona a la que va recibida la llamada. En la segunda comprobamos el estado de la llamada para enviarla al contestador (tercera) o para colgar en caso que no tenga contestador.

```
exten=>333,1,Hangup;
//Cuando alguien llama al 333, la llamada salta con

//prioridad 1 y se colgará al final.
exten=>3000,1,Dial(SIP/3000,30,Ttm);
//Realiza una llamada con permisos de transferencia
exten=>3000,2,Hangup;
//Cuando la llamada acaba se cuelga
exten=>3000,102,VoiceMail(3000);
//Las prioridades 101+ significan que el usuario no estaba

//registrado, con esta hacemos que salte el contestador
exten=>3000,103,Hangup
//colgamos

exten=>_7.,1,Pickup(${EXTEN:1}@default);
//Marcación de un 7 seguido de un número para

//capturarle una llamada (pick-up)
exten=>s,1,Wait(1);
//Espera 1 segundo
exten=>s,2,Answer;
//Asterisk contesta la llamada
exten=>s,3,DigitTimeout(5);
//Timeout de 5 segundos.
exten=>s,4,ResponseTimeout(10);
```

```
//Response timeout de 10 segundos
exten=>s,5,BackGround(test);
//Ejecucion de un mensaje de voz
exten=>s,6,Hangup
exten=>1000,1,Goto(exemple2,s,1);
//Cuando llamamos a 100, la llamada se va hacia 1000
```

Figura 2.22. Ejemplo extensions.conf

Por último, me gustaría comentar un ejemplo muy interesante, es la manera mediante la que yo separo las llamadas para, de manera que según en que contexto coloque a cada extensión, tendrá permisos para realizar un tipo de llamadas u otras.

Mirando la figura de debajo, podemos ver diferentes contextos enlazados, de manera que desde el contexto nivell1 tan sólo vemos las extensiones internas, desde el nivell2 podemos ver las extensiones nacionales, las de atención al usuario y las internas. Siguiendo estos dos ejemplos, si colocamos una extensión con el contexto nivell2, está tendrá permisos para llamar al números fijos nacionales, mientras que si le asignamos el contexto nivell1 a este mismo número no los tendrá.

```
[nivell1]
;home
include => home
;include => default

[nivell2]
;home i trucades nacionals
include => nivell1
include => nacionals
include => atencio

[nivell3]
;home, trucades nacionals i mobils
include => nivell2
include => mobils
include => internacionals

[nivell4]
;home, trucades nacionas, mobils i internacionals
include => nivell3
include => internacionals
```

Figura 2.23. Ejemplo extensions, división de llamadas según destino

El ejemplo anterior me parece uno de las ideas que más nos han simplificado el trabajo de la configuración del dialplan.

2.4.4.4 *agents.conf* y *queues.conf*

Estos ficheros están separados pero su funcionamiento está totalmente vinculado. En uno se definen las colas, que pueden ser gestionadas por agentes definidos en el otro.

Una cola se interpreta como una lista donde dejamos aparcadas las llamadas a la espera de ser atendidas mientras que un agente es una persona que contestará llamadas que vienen de una cola.

Para crear un agente abriremos el archivo *agents.conf*. En el podemos ver que los parámetros que nos piden son básicos. Para cada persona que sea un agente, tendremos que rellenar la siguiente línea:

agent=>Extension,password,Nombre de Agente

Por ejemplo:

```
[home]
agent=>10,1111,Jordi
agent=>12,2222,Juan
```

Figura 2.24. Creación de un agente en Asterisk

Cada agente utilizará un teléfono para recibir llamadas, pero no tiene que ser el mismo. Cuando una operadora llega a su lugar de teléfono, tiene que presionar “#” y después realizar un pickup para ser conectada.

Para crear una cola abriremos el archivo *queues.conf*. En el podemos añadir la siguiente definición para una cola:

```
[colatest]
music=default
strategy=ringall;           //Estrategia de llamadas cuando hay varios agentes .
timeout=15
retry=5
wrapuptime=0
maxlen=0
announce-frequency=no
announce-holdtime=no
member=>Agent/10
member=>Agent/12
```

Figura 2.25. Ejemplo *extensions.conf*

Concretamente. Esta cola llamará a dos agentes a la vez cuando llegue una llamada. Se pueden realizar extensiones directas como agentes, pero no es recomendable. El agente se tiene que registrar en el sistema, de manera que le podamos enviar en todo momento llamadas a la extensión en la que esté dado de alta.

En el ejemplo de abajo, podemos ver dos líneas del extensions.conf que hacen que cuando se llame al 71, pide el usuario y contraseña para logar al agente en la cola mediante el comando de Asterisk AgentCallbackLogin.

Cuando llamamos al 72, se le pedirá otra vez usuario y contraseña del agente. Puede seleccionar otra extensión o apretar # para salir de la cola.

```
;//Login:
exten=>71,1,AgentCallbackLogin(|s|${CALLERID(NUM)})
;//Logout:
exten=>72,1,AgentCallbackLogin(||1)
```

Figura 2.26. Ejemplo extensions

El registro de usuarios es permanente en Asterisk, es decir, si el servidor se reinicia o se apaga, Asterisk es capaz de recuperar el estado de los Agentes y su registro de la cola.

La solución que implementamos para las operadoras de ESADECREAPOLIS se baso en una Shell de unix que llamaba el login, con su pass, de las operadoras. Este se ejecuta al llamar a un número mediante el siguiente fragmento del extensions.conf

```
;Alta i baixa operadora1
exten => 5989,1,System(/usr/local/etc/asterisk/op1_on.sh)
exten => 5989,2,Hangup
exten => 5988,1,System(/usr/local/etc/asterisk/op1_off.sh)
exten => 5988,2,Hangup
```

Figura 2.27. Alta operadoras en extensions.conf

2.4.4.5 *meetme.conf*

Para poder utilizar la función de salas de conferencias hace falta tener compilado el módulo de ztdummy.ko en el sistema. También tenemos que marcar su carga en el archivo de inicio del módulo zaptel de Asterisk (/usr/local/rc.d/zaptel) añadiendo ztdummy.ko en la línea de kmod_load y en la línea kmod_unload.

Para configurar las salas de conferencia editaremos el archivo meetme.conf. En el tan sólo tenemos que añadir la línea “*conf=número*” de extensión de la sala. Como el siguiente ejemplo.

```
conf=601
```

Figura 2.28. Ejemplo configuración meetme

2.4.4.6 *manager.conf*

Este archivo nos permitirá crear usuarios que puedan ejecutar comandos de Asterisk. Por ejemplo, para la administración de Asterisk mediante aplicaciones que no se encuentran en el mismo servidor (frontends de configuración, operadores, tarificadores, etc.), es necesario crear cuentas de manager.

Para ello abrimos el archivo managers.conf y podemos añadir usuarios con el siguiente formato:

```
[manager]
secret=4444
read=system,call,log,verbose,command,agent,user,config
write=system,call,log,verbose,command,agent,user,config
enabled=yes
webenabled=yes
```

Figura 2.29. Configuración de managers en Asterisk

Podemos ver que cada manager tendrá una clave y permisos de escritura/lectura concretos. Así como podemos habilitarle el acceso Web mediante las otras dos opciones.

2.4.4.7 *cdr_custom.conf*

Las llamadas de Asterisk tienen muchos parámetros (Origen, destino, duración, etc.) que hace que ver un log de ellos sea muy poco legible.

Con cdr_custom podemos crear un archivo que contenga una vista rápida de las llamadas realizadas y recibidas y algunos de sus parámetros. De esta manera podremos calcular la tarificación de las llamadas mediante este archivo de configuración.

Por defecto, el archivo con la vista rápida se guardará en /var/log/asterisk/cdr-custom/Master.csv.

Para definir la salida el reporte que obtendremos tenemos que abrir cdr-custom.conf y editar la siguiente línea:

```
[mappings]
Master.csv =>"src:${CDR(src)}","dst:${CDR(dst)}","dcontext:${CDR(dcontext)}",
"start:${CDR(start)}","answer:${CDR(answer)}","end:${CDR(end)}",

"duration:${CDR(duration)}","billsec:${CDR(billsec)}","disposition:${CDR(disposition)}"
```

Figura 2.30. Configuración resumen personalizado de las llamadas**2.4.5 Panel de operadora**

Una vez tenemos los archivos de Asterisk definidos, tenemos que integrar el panel de la operadora con Asterisk.

Para ello empezaremos con la configuración del apache que anteriormente hemos instalado. La configuración por defecto concuerda con lo que necesitamos salvo por un par de cambios:

En primer lugar hace falta modificar la línea del Servername por la IP 192.168.6.12:80, que es la IP del servidor.

Por último, el “document root”, que es la carpeta en la que apache busca los archivos de sus Webs, es la carpeta /usr/local/www/apache22/data donde tendremos que colocar la carpeta de Flash Operator Panel.

Una vez ya tenemos el Apache configurado, empezaremos con la configuración del panel de la operadora. La operadora es una aplicación implementada en Perl que interactúa con Asterisk y mediante una interfaz llamada “variables”, tendremos que asegurar los permisos de lectura y escritura a este archivo, consigue describir el estado en el que se encuentran las líneas de Asterisk.

Para configurarlo tendremos que modificar tres archivos ubicados en “/usr/local/www/op_panel”; op_server.cfg, op_buttons.cfg y op_style.cfg.

2.4.5.1 op_server.cfg

En este archivo encontramos las opciones generales de la aplicación. Tenemos que configurarlo para que FOP pueda comunicarse con Asterisk mediante el manager que hemos definido en Asterisk. También tenemos que especificar donde se encuentra el directorio de flash.

Todo esto lo podemos definir de la siguiente manera:

```
[general]
manager_host=192.168.6,12
manager_port=5038
manager_user=manager
manager_secret=4444
auth_md5=1
astmanproxy_port=5038
astmanproxy_server=192.168,6,12
listen_port=4445
flash_dir=/usr/local/www/apache22/data/fop/ security_code=5069
```

Figura 2.31. Configuración Asterisk en FOP

2.4.5.2 *op_style.cfg*

En este archivo podemos definir el libro de estilos de la aplicación, con lo que mediante él podemos cambiar toda la parte visual del mismo.

La configuración por defecto de este archivo es muy válida, sin embargo, si tenemos que añadir muchas extensiones si tendremos que modificarlo, dado que está pensado para un número determinado de teléfonos.

2.4.5.3 *op_buttons.cfg*

Este es el archivo donde definiremos los teléfonos que queramos que queden monitorizados en FOP, esto lo traduciremos en los botones de la aplicación.

En este archivo podemos definir tres tipos de botones según a que ente se refieren; a extensiones, a colas y a salas de conferencias.

Para definir una extensión SIP, se necesita realizar una configuración como la que muestro, es obvio que los parámetros tendrán que concordar con los del archivo sip.conf de Asterisk:

```
[SIP/5077]
Position=2           //El estilo marca las posiciones
Label="Jordi"        //Label que aparecerá en la Web
Server=1
Extension=5077
Context=nivell3
Icon=1
```

Figura 2.32. Configuración extensión FOP

Para definir las colas y las salas de conferencias lo haremos de la siguiente

manera:

```
Cua
[QUEUE/colatest]
Position=10-12
Label="Operadora test"
Extension=71
Icon=3

Sala conferència
[601]
Position=23
Label="Conferencia"
Extension=601
Context=home
Icon=5
```

Figura 2.33. Configuración cola FOP

Con tal de tener una vista más agradable, podemos definir separaciones y colores mediante rectángulos y leyendas de la siguiente manera:

```
[rectangle]
x=748
y=35
width=252
height=220
line_width=3
line_color=ff1010
fade_color1=ff1010
fade_color2=a01000
rnd_border=2
alpha=20
layer=top
panel_context=*

[LEGEND]
x=790
y=50
text=Conferences
font_size=32
font_color=FF0000
font_family=Arial
use_embed_fonts=
```

Figura 2.34. Configuración colores FOP

2.5 Integración en ESADECREAPOLIS

Una vez tenemos Asterisk instalado y configurado, lo implantaremos en las infraestructuras del edificio. Con este proceso culminaremos un servicio y estará listo para utilizarse.

2.5.1 Descripción de la red

El edificio tiene una forma rectangular, sobre unos 200 metros de largo por tan sólo 30 metros de ancho y 5 plantas de altura por dos de sótano. A partir de la segunda planta, las tres restantes se dividen en cuatro módulos independientes.

El núcleo del cableado horizontal está en la primera planta dividido en cuatro armarios repartidos a lo largo del edificio, el cableado vertical sale de cada uno de los armarios para llegar al punto final de usuario en las dos primeras plantas, mientras que para cada una de las tres plantas restantes se reparte en pequeños armarios de pared para cada uno de los cuatro módulos previamente comentados,

Por otro lado, en la segunda planta del sótano hay ubicado un CPD en el que se encuentran los servidores de ESADECREAPOLIS y de algunos de los residentes.

Por último, hay una habitación reservada para que los operadores puedan entregar el servicio, la sala de operadores se encuentra en la planta 0.

En cuanto al cableado del edificio, todo es cable UTP de comunicaciones de datos certificado en clase seis de la marca Systimax excepto en el cableado horizontal de la primera planta, donde los Ritis se comunican entre ellos mediante fibra óptica.

Todo el switching se realiza mediante equipos Enterasys C3 con bocas a Gigabit con PoE, mientras que el enrutado se basa en equipos Cisco Catalyst 3750. Ambos equipos tienen tarjetas de fibra a diez gigabits por segundo.

2.5.2 Integración de Asterisk

El modelo final queda como podemos ver en la siguiente figura. El objetivo final es colocar Asterisk en una zona física y en una lógica que, por un lado, le permita trabajar sin problemas de una manera eficiente y por otro, las máquinas se encuentren en un entorno donde no haya altas temperaturas ni una humedad relativa alta, es decir, donde puedan trabajar con garantías.

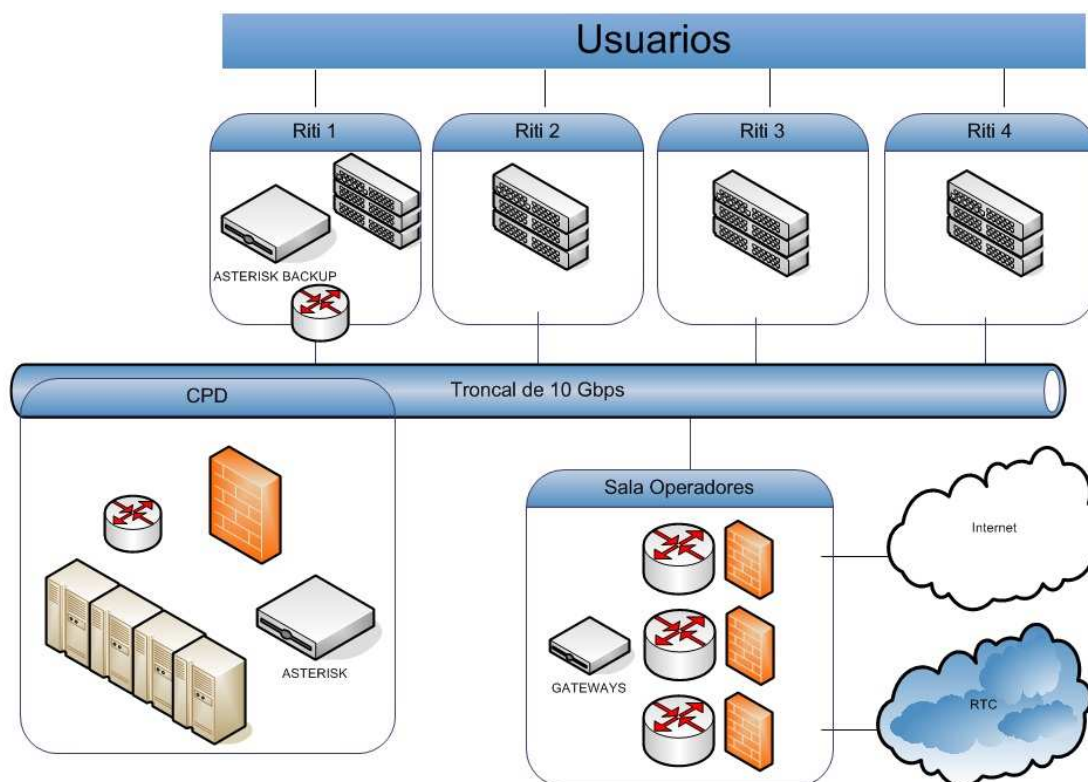


Figura 2.35.. Red telefónica de ESADECREAPOLIS

En ella podemos ver una representación de lo que previamente he explicado, vemos los cuatro armarios de comunicaciones unidos mediante la troncal de datos con el CPD y la sala de operadores, que es la que enlaza con Internet y con la red telefónica conmutada.

La primera cuestión que se planteó fue donde colocar los servidores Asterisk, la decisión fue colocar los dos servidores en lugares distintos, por tal de prevenir catástrofes o cortes puntuales. Como podemos ver en la imagen anterior, los sitios que elegimos fueron el CPD para el servidor que estará operativo y el RITI1 para el servidor de backup. La elección del CPD para el servidor principal era obvia, sin embargo la ubicación del segundo servidor presentaba dudas. Al final, las buenas condiciones del ambiente y el control del acceso nos llevaron a escoger el Riti1 como lugar para colocar el segundo servidor.

Lógicamente, ambos servidores comparten una VLAN de trabajo así como un rango de IPs donde tienen plena visibilidad de los servicios de copias de seguridad y los otros servidores de servicios. Utilizaremos la misma NAS que utilizamos para realizar copias de seguridad de los otros servidores para complementar las de Asterisk, de esta manera, si algún día tenemos algún problema podremos restaurar una copia antigua de Asterisk. También cabe comentar que ambos servidores están sincronizados con una periodicidad diaria.

Los Gateways están colocados en la sala de operadores, donde el operador nos entrega el servicio, estos se enlazan a primarios de treinta líneas si van a

la red telefónica conmutada o a la conexión a Internet si van dirigidas a un operador de Internet.

Pasamos al usuario final, los teléfonos IP se alimentarán mediante la electrónica de red, que es PoE (Power Over Ethernet) por lo que no nos tenemos que preocupar. Todos estos teléfonos trabajarán en la misma VLAN, que será dedicada para este servicio. Esto nos permitirá asignar calidad de servicio en la red para la telefonía por delante de otros tráficos.

Una vez tenemos el teléfono encendido, el protocolo Cisco Discovery Protocol, que lo soporta la electrónica del edificio, reconocerá el teléfono y le asignará una VLAN de gestión telefónica. Entonces el teléfono cogerá una IP del rango de telefonía que le asignará el enrutador por defecto de su VLAN. Como la VLAN es dedicada, no tendremos ningún otro DHCP en la red que nos moleste. El mismo DHCP le informará de la dirección IP de un servidor TFTP. Este servidor tiene principalmente dos funciones; por un lado permite actualizaciones de firmware de los teléfonos (algo muy importante, dado que por defecto los CISCO no contienen SIP) y asignar toda la configuración del teléfono según su Mac. Con esto conseguimos que los teléfonos, estén donde estén colocados, siempre tengan la misma extensión telefónica y estén asignados a la misma persona.

Con estos pasos ya tenemos un teléfono funcional en cada puesto de trabajo de ESADECREAPOLIS.

2.6 Resultados de la telefonía

En líneas generales, la telefonía ha tenido una muy buena aceptación. Puedo decir que en torno al 60 por ciento de todos los inquilinos del parque lo utilizan asiduamente y tienen una opinión favorable.

Pese a que no son conceptos nuevos, el gran número de emprendedores y pequeñas empresas que hay en el parque hace que muchos de los usuarios se sorprendan de las funcionalidades que presenta la centralita. Otros casos, como los clientes que habían trabajado con telefonía IP, no han comentado que les faltase ninguna funcionalidad que tuvieran en otra centralita. Y por último, hay casos muy puntuales de inquilinos que ya tenían su centralita IP y la han instalado en su oficina, en este caso todos nos pasan las llamadas externas a nuestro Asterisk para salir por nuestros primarios.

Si nos fijamos en cifras más significativas. El número de extensiones no ha parado de crecer proporcionalmente a la ocupación del edificio, incluso más, dado que posteriormente otro edificio del campus se unió a la telefonía, añadiendo 180 terminales a finales del 2010. No hemos registrado ninguna baja del servicio, tan sólo las derivadas a la baja como inquilino de algún cliente. Actualmente, tenemos un total de 435 teléfonos trabajando cursando llamadas con nuestra centralita, un quinientos por ciento más que los ochenta y dos teléfonos que instalamos de entrada.

En la figura de abajo podemos ver un gráfico que nos muestra el número de extensiones en los últimos dos años.



Figura 2.36. Teléfonos operativos en la centralita.

Por otra parte el consumo telefónico, pese presentarse con altibajos (vacaciones, puntas de trabajo, etc.) muestra una media ascendente igual, aunque con menos pendiente, que la gráfica de aumento de teléfonos. En líneas generales, podemos ver que el consumo final es aproximadamente tres veces más grande que el consumo inicial. Si tenemos en cuenta que tenemos cinco veces más terminales el consumo no ha subido todo lo esperado.

En primer lugar, una consumo medio por cliente se ha apalancado incluso en algunos casos ha bajado un poco debido a la conciencia general en contener gastos que tienen las empresas este año.

Por otro lado, el aumento brusco de teléfonos en el periodo de diciembre del 2009 no se reflejan en el consumo dado al tipo de usuario al que van destinado. El usuario final de estos teléfonos eran estudiantes, estos no tienen un potencial de consumo como el de las empresas, su actividad es mucho menor.

La figura inferior representa el consumo telefónico anual de los dos últimos cursos.

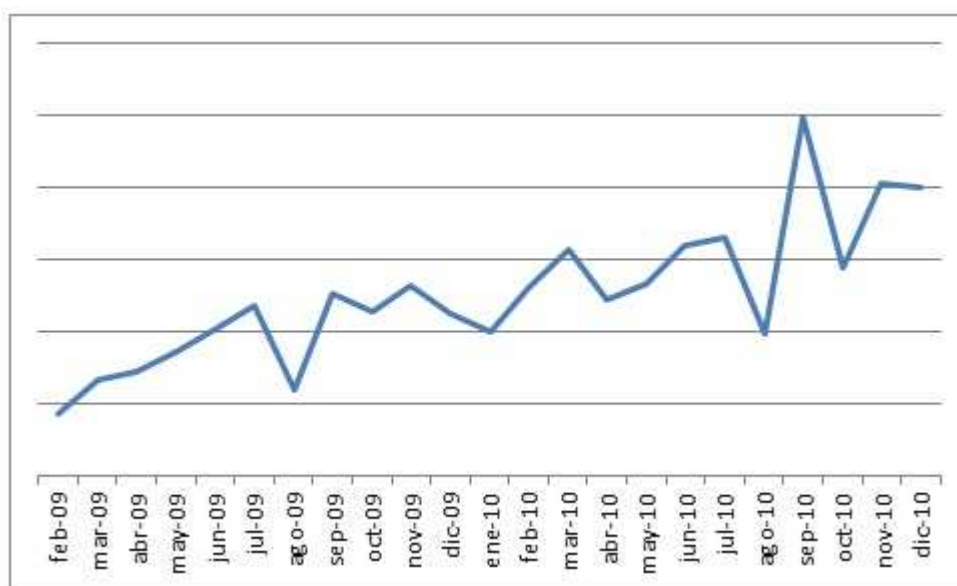


Figura 2.37. Consumo telefónico

La previsión de este próximo 2011 es seguir la evolución mostrada hasta ahora tanto en el consumo como en los terminales instalados.

Bibliografía

- Carpenter C, *Asterisk 1.4*, Packt Publ
- Simionovich N, *Asterisk Gateway Interface Programming*, Packt Publ
- Himanshu D, *Hacking VoIP*, No starch Press
- Van Meggelen J, *The Future of telephony*, O'reilly
- Ransome J F, *VoIP Security*, Elsevier digital Press
- Wikipedia



Escola Politècnica Superior
de Castelldefels

UNIVERSITAT POLITÈCNICA DE CATALUNYA

ANNEXOS

TÍTOL DEL TFC: Implantación de una centralita VoIP basada en Asterisk en un parque de innovación

TITULACIÓ: Enginyeria Tècnica de Telecomunicació, especialitat Telemàtica

AUTOR: Jordi Piñol Casas

DIRECTOR extern: Javier Fernandez Garcia

DIRECTOR: Juan López Rubio

DATA: 5 de Gener del 2011

Annexos

SIP.conf (Dos columnas)

```
[general]
context=incoming
bindport=5060
bindaddr=0.0.0.0
srvlookup=yes
disallow=all
allow=h263
allow=ulaw
allow=alaw
videosupport=yes
nat=yes
canreinvite=no
localnet=x
localnet=x
externip=x
language=ca
dtmfmode=rfc2833
deny=0.0.0.0/0.0.0.0
permit=x
permit=x
```

[authentication]

```
;gateway Esade
[192.169.48.x]
type=friend
host=192.169.48.x
canreinvite=no
nat=no
disallow=all
allow=g729
context=incoming
languaje=en
dtmfmode=rfc2833
```

;Gateway

```
[192.169.48.x]
type=friend
host=192.169.48.x
canreinvite=no
nat=no
disallow=all
allow=g729
```

```
context=incoming
languaje=en
dtmfmode=rfc2833
```

[5788]

```
type=friend
host=192.169.48.6
callerid=5788
qualify=yes
canreinvite=no
disallow=all
allow=all
;allow=ulaw
context=miratel
language=en
dtmfmode=rfc2833
```

[5789]

```
type=friend
host=192.169.48.6
callerid=5788
qualify=yes
canreinvite=no
disallow=all
allow=all
;allow=ulaw
context=miratel
language=en
dtmfmode=rfc2833
```

[5790]

```
type=friend
host=192.169.48.6
callerid=5788
qualify=yes
canreinvite=no
disallow=all
allow=all
;allow=ulaw
context=miratel
language=en
dtmfmode=rfc2833
```

El SIP,conf sigue con un registro para cada teléfono hasta llegar a los 345 que tenemos operativos en este momento.

Voicemail.conf

```
[general]
format = wav49|gsm|wav
serveremail = voicemail@esadecreapolis.com
attach = yes
skipms = 3000
maxsilence = 10
silencethreshold = 128
maxlogins = 3
maxmsg = 5000
externnotify = /usr/local/etc/asterisk/voicemailperm.sh
emailsubject = Creapolis Voicemail: New message ${VM_MSGNUM} in mailbox
${VM_MAILBOX}
emailbody = Dear ${VM_NAME}:\n\n\tjust wanted to let you know you were just
left a ${VM_DUR} long message (number ${VM_MSGNUM})\nin mailbox
${VM_MAILBOX} from ${VM_CALLERID}, on ${VM_DATE}, so you
might\nwant to check it when you get a chance. Thanks!\n\n\t\t\t\t\t--IT Support
Services.\n\n\n\n\n\n Please, do not reply this message.
pagerfromstring = Voicemail
pagersubject = New VM
pagerbody = New ${VM_DUR} long msg in box ${VM_MAILBOX}\n from
${VM_CALLERID}, on ${VM_DATE}
emaildateformat = %A, %B %d, %Y at %r
mailcmd = /usr/sbin/sendmail -v -t -f voicemail@esadecreapolis.com
attach = yes
sendvoicemail = yes ; Context to Send voicemail from [option 5 from the
advanced menu]
```

```
[zonemessages]
eastern = America/New_York|'vm-received' Q 'digits/at' IMp
central = America/Chicago|'vm-received' Q 'digits/at' IMp
central24 = America/Chicago|'vm-received' q 'digits/at' H N 'hours'
military = Zulu|'vm-received' q 'digits/at' H N 'hours' 'phonetic/z_p'
```

```
[default]
```

```
[other]
```

```
[voicemail]
5000 => 1234,Recepcio,recepcio@clubcreapolis.com
```

Extensions.conf

```
[general]
static=yes
```

```
writeprotect=no
autofallthrough=yes
clearglobalvars=no
priorityjumping=no
```

```
[globals]
CONSOLE=Console/dsp
IAXINFO=guest
TRUNK=Zap/g2
TRUNKMSD=1
```

```
;Include residencia
#include "extensionsresidenciamacro.conf"
```

```
[macro-voicemail1]
exten => s,1,Dial(${ARG1},20)
exten => s,2,Goto(s-${DIALSTATUS},1)
exten => s-NOANSWER,1,VoiceMail(su${MACRO_EXTEN}@voicemail)
exten => s-NOANSWER,2,Hangup()
exten => s-BUSY,1,VoiceMail(sb${MACRO_EXTEN}@voicemail)
exten => s-BUSY,2,Hangup()
exten => _s.,1,Goto(s-NOANSWER,1)
```

```
[macro-voicemail1-rec]
exten => s,1,Set(CALLFILENAME=${ARG1}-
${STRFTIME(${EPOCH},GMT+8,%C%y%m%d%H%M))
exten => s,2,Monitor(wav,${CALLFILENAME},m)
exten => s,3,Dial(${ARG1},20,W)
exten => s,4,Goto(s-${DIALSTATUS},1)
exten => s-NOANSWER,1,VoiceMail(su${MACRO_EXTEN}@voicemail)
exten => s-NOANSWER,2,Hangup()
exten => s-BUSY,1,VoiceMail(sb${MACRO_EXTEN}@voicemail)
exten => s-BUSY,2,Hangup()
exten => _s.,1,Goto(s-NOANSWER,1)
```

```
[macro-voicemailrecepcio]
exten => s,1,Dial(SIP/5009,20)
exten => s,2,Goto(s-${DIALSTATUS},1)
exten => s-NOANSWER,1,VoiceMail(su5002@voicemail)
exten => s-NOANSWER,2,Hangup()
exten => s-BUSY,1,VoiceMail(sb5002@voicemail)
exten => s-BUSY,2,Hangup()
exten => _s.,1,Goto(s-NOANSWER,1)
```

```
;;;;;;;;;;;;;Incoming
[incoming]
```

```
;;;;; Importem totes les redireccions de la residencia
#include "extensionsresidencia.conf"
```

```

exten => _85075,1,Answer()
exten => _85075,2,Set(CALLERID(num)=86${CALLERID(num)})
exten => _85075,3,Goto(home,5075,1)
exten => _850XX,1,Answer()
exten => _850XX,2,Set(CALLERID(num)=86${CALLERID(num)})
exten => _850XX,3,Dial(SIP/${EXTEN:1})
exten => _857XX,1,Answer()
exten => _857XX,2,Set(CALLERID(num)=86${CALLERID(num)})
exten => _857XX,3,Dial(SIP/${EXTEN:1})
exten => _88XXX,1,Answer()
exten => _88XXX,2,Set(CALLERID(num)=86${CALLERID(num)})
exten => _88XXX,3,Dial(SIP/${EXTEN:1})
exten => _82XXX,1,Set(CALLERID(num)=86${CALLERID(num)})
exten => _82XXX,2,Dial(ooh323/${EXTEN:1}@192.169.30.1:1720)
exten => _83XXX,1,Set(CALLERID(num)=86${CALLERID(num)})
exten => _83XXX,2,Dial(ooh323/${EXTEN:1}@192.169.30.1:1720)
exten => _84XXX,1,Set(CALLERID(num)=86${CALLERID(num)})
exten => _84XXX,2,Dial(ooh323/${EXTEN:1}@192.169.30.1:1720)
exten => _85XXX,1,Set(CALLERID(num)=86${CALLERID(num)})
exten => _85XXX,2,Dial(ooh323/${EXTEN:1}@192.169.30.1:1720)
exten => 935543000,1,SetCallerID(Creapolis <0${CALLERID(NUM)}>)
exten => 935543000,2,Goto(home,5000,1)
;exten => 935543000,2,Goto(ivrrepcion,s,1)
;exten => 935878636,4,Hangup()
;
;
;
; Recortado, estan todos los DDIs
;
;
exten => s,1,Hangup()
exten => _X,1,Hangup()
exten => _X.,1,Hangup()
exten => i,1,Hangup()
exten => t,1,Hangup()

```

```

.....Default

```

```

[default]

```

```

include => incoming

```

```

[grabar]

```

```

exten => 205,1,Answer
exten => 205,2,Wait(2)
exten => 205,3,Record(GRABACIO%d:gsm)
exten => 205,4,Wait(2)
exten => 205,5,Playback(${RECORDED_FILE})
exten => 205,6,Wait(2)
exten => 205,7,Hangup

```

```

[conference1]

```

```

.....Sala Conferencia 600

```



```
exten => 600,1,MeetMeCount(1234,CONF_COUNT)
exten => 600,2,GotoIf($[${CONF_COUNT}<=8]?3:100)
exten => 600,3,MeetMe(1234,DpM,3)
exten => 600,100,Playback(conf-full)
```

[ivrrepcio]

```
exten => s,1,set(TIMEOUT(Digit)=5)
exten => s,2,set(TIMEOUT(Response)=20)
exten => s,3,Answer()
exten => s,4,Ringing(5)
exten => s,5,Wait(3)
```

```
exten => s,6,SetCallerID(MarcoPolo <0${CALLERID(NUM)}>)
```

```
exten => s,7,GotoIf("${CALLERID(name)}" = "Creapolis"?200)
exten => s,8,GotoIf("${CALLERID(name)}" = "Brandnew"?202)
exten => s,9,GotoIf("${CALLERID(name)}" = "MarcoPolo"?204)
exten => s,10,GotoIf("${CALLERID(name)}" = "Inbens"?206)
exten => s,12,GotoIf("${CALLERID(name)}" = "Kiminkgest"?208)
exten => s,13,Goto(ivrrepcio,s,300)
```

```
exten => s,200,Background(esadecreapolis-welcome)
exten => s,201,Goto(ivrrepcio,s,300)
exten => s,202,Background(brandnew-welcome)
exten => s,203,Goto(ivrrepcio,s,300)
exten => s,204,Background(marcopolo-welcome)
exten => s,205,Goto(ivrrepcio,s,300)
exten => s,206,Background(inbens-welcome)
exten => s,207,Goto(ivrrepcio,s,300)
exten => s,208,Background(kiminkgest-welcome)
exten => s,209,Goto(ivrrepcio,s,300)
```

```
exten => s,300,Queue(operadora,tTr,,,10)
exten => s,301,Background(esadecreapolis-busylines-es)
exten => s,302,Background(esadecreapolis-busylines-en)
exten => s,303,WaitExten(2)
exten => s,304,Goto(ivrrepcio,s,300)
```

```
exten => 1,1,VoiceMail(su5009@voicemail)
exten => 1,2,Hangup
```

```
exten => _X.,1,Goto(ivrrepcio,s,300)
exten => i,1,Goto(ivrrepcio,s,300)
exten => t,1,Goto(ivrrepcio,s,300)
```

[ivrepresa]

```
exten => s,1,set(TIMEOUT(Digit)=2)
exten => s,2,set(TIMEOUT(Response)=20)
exten => s,3,Answer()
```

```

;Missatge Bienvenido a Flowlab
exten => s,4,Playback(flowlab-welcome)
;exten => s,5,wait(1)
exten => s,5,GotoIf("${CALLERID(NUM)}"="913983868"?10)
;Missatge Si quiere hablar con Administracion pulse 1
;exten => s,5,Background(flowlab-administracion)
;Missatge Con Comercial pulse 2
;exten => s,6,Background(flowlab-comercial)
;Missatge Con Marketing pulse 3
;exten => s,7,Background(flowlab-marketing)
;Missatge Con Produccion pulse 4
;exten => s,8,Background(flowlab-produccion)
;Missatge Con RRHH pulse 5
;exten => s,9,Background(flowlab-RRHH)
;exten => s,10,WaitExten(4)
;Redirigim a Cua Flowlab si en 4 segons no marquen res
exten => s,6,Dial(SIP/5778,10)
exten => s,7,Dial(SIP/5775,10)
exten => s,8,Queue(flowlab,tTr,,,16)
exten => s,9,VoiceMail(su5775@voicemail)
exten => s,10,Hangup

```

```

;exten => 1,1,Set(CALLERID(all)=Administracio <5775>)
;exten => 1,2,Goto(mobils,0935898022,1)
;exten => 2,1,Set(CALLERID(all)=Comercial <5775>)
;exten => 2,2,Goto(mobils,0650452769,1)
;exten => 3,1,Set(CALLERID(all)=Marketing <5775>)
;exten => 3,2,Goto(mobils,0671091127,1)
;exten => 4,1,Set(CALLERID(all)=Produccio <5775>)
;exten => 4,2,Goto(mobils,0637302147,1)
;exten => 5,1,Set(CALLERID(all)=RRHH <5775>)
;exten => 5,2,Goto(mobils,0663196096,1)

```

```

exten => _X.,1,Goto(ivflowlab,s,6)
exten => i,1,Goto(ivrflowlab,s,6)
exten => t,1,Goto(ivrflowlab,s,6)

```

```

;
;
;

```

```

;Recortadas IVRs, datos personales de empresas.

```

```

;
;
;

```

```

;Missatge benvinguda4

```

```

exten => 5,2,Goto(mobils,0663196096,1)

```

```

;Tenir en compte la data del sistema. Realment es de 9 a 17:30

```

```

exten => s,5,Background(greenlab-intro)

```

```

exten => s,6,Wait(1)

```

```

exten => s,7,GotoIfTime(9:00-17:30|mon-fri|*|*?ivrgreenlab,s,20)

```

```

;Missatge Fora hora

```

```
exten => s,8,Background(greenlab-forahorari)
exten => s,9,VoiceMail(s5740@voicemail)
exten => s,10,Hangup
```

```
;Missatge Vacances
exten => s,15,Background(greenlab-vacances)
exten => s,16,VoiceMail(s5740@voicemail)
exten => s,17,Hangup
```

```
;Comprobacio si es divendres
exten => s,20,GotoIfTime(14:00-17:30|fri|*|*?ivrgreenlab,s,8)
```

```
;Missatge si quiere hablar con
exten => s,21,Background(greenlab-numeros)
;Redirigim a Telefon de la Rebeca si en 5 segons no marquen res
exten => s,22,WaitExten(5)
exten => s,23,Goto(home,5740,1)
exten => s,24,Hangup
```

```
exten => 1,1,Set(CALLERID(all)=To Rebeca <5740>)
exten => 1,2,Goto(home,5740,1)
exten => 2,1,Set(CALLERID(all)=To Francesc <5741>)
exten => 2,2,Goto(home,5741,1)
exten => 3,1,Set(CALLERID(all)=To Olga <5742>)
exten => 3,2,Goto(home,5742,1)
exten => 4,1,Set(CALLERID(all)=To Albert <5084>)
exten => 4,2,Goto(home,5084,1)
```

```
exten => _X.,1,Goto(home,5740,1)
exten => i,1,Goto(home,5740,1)
exten => t,1,Goto(home,5740,1)
```

```
[ivralihoop]
exten => s,1,set(TIMEOUT(Digit)=2)
exten => s,2,set(TIMEOUT(Response)=20)
exten => s,3,Answer()
```

```
exten => s,4,GotoIfTime(10:00-13:30|mon-fri|*|*?ivralihoop,s,30)
exten => s,5,GotoIfTime(15:00-19:00|mon-fri|*|*?ivralihoop,s,30)
exten => s,6,Wait(1)
```

```
;Missatge Fora hora
exten => s,7,Background(alihoop-forahorari-cast)
exten => s,8,Background(alihoop-forahorari-cat)
exten => s,9,VoiceMail(s5727@voicemail)
exten => s,10,Hangup
```

```
exten => s,30,Background(alihoop-intro-cast)
exten => s,31,Background(alihoop-intro-cat)
exten => s,32,Queue(alihoop,t,,,120)
```

```
exten => s,33,Background(alihoop-foracua-cast)
exten => s,34,Background(alihoop-foracua-cat)
exten => s,35,VoiceMail(s5727@voicemail)
exten => s,36,Hangup
```

```
exten => _X.,1,Goto(ivralihoop,s,1)
exten => i,1,Goto(ivralihoop,s,1)
exten => t,1,Goto(ivralihoop,s,1)
```

```
.....
redirecciones de numerous internos
.....
```

[home]

```
.....Cua Operadora
.....
```

```
;exten => 5000,1,Goto(home,5009,1)
```

```
exten => 5000,1,Goto(home,5998,1)
;exten => 5000,1,Goto(ivrrecepicio,s,1)
exten => 5998,1,Queue(operadora,tTr,,,30)
exten => 5998,2,VoiceMail(su5009@voicemail)
exten => 5998,3,Hangup
```

```
;exten => 5964,1,Goto(home,5974,1)
;Menta
exten => 5971,1,Dial(SIP/5971,18)
exten => 5971,2,goto(home,5964,1)
exten => 5972,1,Dial(SIP/5972,18)
exten => 5972,2,goto(home,5964,1)
```

```
;exten => 5005,1,Goto(ivrtic,s,1)
exten => 5005,1,Queue(it,tTr,,,40)
```

```
;Alta i baixa operadora1
exten => 5989,1,System(/usr/local/etc/asterisk/op1_on.sh)
exten => 5989,2,Hangup
exten => 5988,1,System(/usr/local/etc/asterisk/op1_off.sh)
exten => 5988,2,Hangup
;Alta i baixa operadora2
exten => 5987,1,System(/usr/local/etc/asterisk/op2_on.sh)
exten => 5987,2,Hangup
exten => 5986,1,System(/usr/local/etc/asterisk/op2_off.sh)
exten => 5986,2,Hangup
;Alta i baixa operadora3
exten => 5985,1,System(/usr/local/etc/asterisk/op3_on.sh)
exten => 5985,2,Hangup
exten => 5984,1,System(/usr/local/etc/asterisk/op3_off.sh)
exten => 5984,2,Hangup
```

```
exten => 5983,1,Goto(ivrgreenlab,s,1)
exten => 5982,1,Goto(ivrepilon,s,1)
exten => 5981,1,Goto(ivralihoop,s,1)
```

```
.....;Cua flowlab
;.....
exten => 5996,1,Dial(SIP/5778,10)
exten => 5996,2,Dial(SIP/5775,10)
exten => 5996,3,Queue(flowlab,tTr,,40)
exten => 5996,4,VoiceMail(su5775@voicemail)
exten => 5996,5,Hangup
```

```
.....;Voicemail
;.....
exten => 5999,1,VoiceMailMain(${CALLERID(NUM)})@voicemail)
exten => 5999,n,HangUp()
```

```
.....;Conference
;.....
exten => 5994,1,Goto(conference1,600,1)
;exten => 5994,1,Playback(tonteria)
```

```
.....;Extensions preassignades
;.....
;exten => 5001,1,Macro(voicemail1,SIP/5002)
exten => 5002,1,Macro(voicemail1,SIP/5002)
exten => 2384,1,Macro(voicemail1,SIP/5001)
exten => 5075,1,Dial(SIP/5059&SIP/5079,8)
exten => 5075,2,Ringing(1)
exten => 5075,3,Dial(SIP/0607819512@192.169.48.9)
```

```
exten => 5019,1,Dial(SIP/5019,13)
;exten => 5019,2,Ringing(1)
;exten => 5019,2,Dial(SIP/0699638442@192.169.48.9)
```

```
exten => 5971,1,Dial(SIP/5971,10)
exten => 5971,2,goto(home,5964,1)
```

```
exten => 5972,1,Dial(SIP/5971,10)
exten => 5972,2,goto(home,5964,1)
```

```
;exten => 5028,1,set(sip_CODEC=h263)
exten => 5028,1,Dial(SIP/5028@192.169.49.47)
;exten => 5029,1,set(sip_CODEC=h263)
exten => 5029,1,Dial(SIP/5029@192.169.49.46)
```

```
.....;Grabacio missatges
;.....
exten => 5995,1,Goto(grabar,205,1)
;exten => 5995,1,Playback(GRABACIO1.gsm)
exten => 5995,2,Wait(5)
exten => 5995,3,Playback(epsilon_music_02)
```

```
.....
;.....
```

```
;exten => 5997,1,ExtenSpy(5054@nivell3,q)
```

```
;;;;;Extensions telefons
```

```
exten => 5700,1,Dial(SIP/5700,20)
```

```
;exten => 5700,2,Dial(SIP/5702,20)
```

```
exten => 5700,2,VoiceMail(su5700@voicemail)
```

```
exten => 5740,1,Macro(voicemail1,SIP/${EXTEN})
```

```
;;;;;PIGMALIO Baixa
```

```
;exten => 5706,1,Dial(SIP/5706,20)
```

```
;exten => 5706,2,Queue(operadora,tTr,,,20)
```

```
;exten => 5706,3,VoiceMail(su5706@voicemail)
```

```
;;;;;Payma
```

```
;exten => 5040,1,Goto(nacionals,0935752834,1)
```

```
;;;;;Loop
```

```
exten => 5041,1,Goto(nacionals,0933030299,1)
```

```
;;;;;Mira - Cap al seu CM
```

```
exten => 5788,1,Dial(SIP/5788@192.169.48.6)
```

```
exten => 5789,1,Dial(SIP/5789@192.169.48.6)
```

```
exten => 5790,1,Dial(SIP/5790@192.169.48.6)
```

```
exten => 5791,1,Dial(SIP/5791@192.169.48.6)
```

```
exten => 5792,1,Dial(SIP/5792@192.169.48.6)
```

```
exten => 5793,1,Dial(SIP/5793@192.169.48.6)
```

```
exten => 5794,1,Dial(SIP/5794@192.169.48.6)
```

```
;;;;; Generiques.
```

```
exten => _5054,1,Macro(voicemail1-rec,SIP/${EXTEN})
```

```
exten => _50XX,1,Macro(voicemail1,SIP/${EXTEN})
```

```
exten => _57XX,1,Macro(voicemail1,SIP/${EXTEN})
```

```
exten => _59XX,1,Macro(voicemail1,SIP/${EXTEN})
```

```
exten => _8XXX,1,Macro(voicemail1,SIP/${EXTEN})
```

```
exten => _2XXX,1,Dial(ooh323/${EXTEN}@192.169.30.1:1720)
```

```
exten => _3XXX,1,Dial(ooh323/${EXTEN}@192.169.30.1:1720)
```

```
exten => _4XXX,1,Dial(ooh323/${EXTEN}@192.169.30.1:1720)
```

```
exten => _5XXX,1,Dial(ooh323/${EXTEN}@192.169.30.1:1720)
```

```
exten => _7XXX,1,Dial(ooh323/${EXTEN}@192.169.30.1:1720)
```

```
exten => _86XXX,1,Answer()
```

```
;exten => _86XXX,2,Playback(cluster-vacances)
```

```
exten => _86XXX,2,set(sip_CODEC=g729)
```

```
exten => _86XXX,3,Set(CALLERID(num)=8${CALLERID(num)})
```

```
exten => _86XXX,4,Dial(SIP/${EXTEN:2}@gatewayurl)
```

```
exten => _88XXX,1,Goto(home,${EXTEN:1},1)
```

```
;;;;;Redireccions mobils
```

```

;
;
;RECORTADO, datos personales empresas.
;
;

.....:Voicemail
;.....;
exten => _1XXXX,1,VoiceMailMain(${EXTEN:1}@voicemail)
;.....:Pickup
;.....;
exten => _7XXXX,1,Pickup(${EXTEN:1})

;.....:Cues. Login 2001; Logoff 2002
;.....;
;exten => 2001,1,AgentCallbackLogin(|s,${CALLERIDNUM}@home)
;exten => 2002,1,AgentCallbackLogin(||1)

.....:Mallorca
;.....;
exten => _8910,1,Dial(SIP/85@192.168.91.50)
exten => _891X,1,Dial(SIP/${EXTEN:2}@192.168.91.50)

;.....:
;.....;
;.....:Outgoing
;.....;
;.....;
;.....;
[internacionales]
;exten =>
_0034.,1,ExecIf("${CALLERID(num)}"="5005",Set,CALLERID(ANI)=84.88.228.1
39)
;exten => _0034.,2,Dial(SIP/${EXTEN}@89.202.174.188)
;exten => _0042.,1,Dial(SIP/${EXTEN}@89.202.174.188)
exten => _000.,1,Dial(SIP/${EXTEN}@192.169.48.10)
exten => _000.,2,Congestion()
exten => _000.,102,Congestion()

[nacionales]
exten => _08XXXXXXXXX,1,Dial(SIP/${EXTEN}@192.169.48.10)
exten => _08XXXXXXXXX,2,Congestion()
exten => _08XXXXXXXXX,102,Congestion()
exten => 0936222056,1,Dial(SIP/0936222056@192.169.30.251)
exten => 0936222189,1,Dial(SIP/0936222189@192.169.30.251)
exten => 0935572395,1,Dial(SIP/935572395)
exten => 0935572394,1,Dial(SIP/935572394)

exten => _09XXXXXXXXX,1,ExecIf($[ "${CALLERID(num):0:1}"!="5"
],Set,CALLERID(all)=Creapolis <5000>)
exten => _09XXXXXXXXX,2,Dial(SIP/${EXTEN}@192.169.48.10)
exten => _09XXXXXXXXX,3,Congestion()
exten => _09XXXXXXXXX,103,Congestion()
exten => _891XXXXXXXX,1,Dial(SIP/${EXTEN:1}@89.202.174.188)
exten => _892XXXXXXXX,1,Dial(SIP/${EXTEN:1}@89.202.174.188)
exten => _893XXXXXXXX,1,Dial(SIP/${EXTEN:1}@89.202.174.188)
exten => _894XXXXXXXX,1,Dial(SIP/${EXTEN:1}@89.202.174.188)
exten => _895XXXXXXXX,1,Dial(SIP/${EXTEN:1}@89.202.174.188)

```

```
exten => _896XXXXXXXX,1,Dial(SIP/${EXTEN:1}@89.202.174.188)
exten => _897XXXXXXXX,1,Dial(SIP/${EXTEN:1}@89.202.174.188)
exten => _898XXXXXXXX,1,Dial(SIP/${EXTEN:1}@89.202.174.188)
exten => _899XXXXXXXX,1,Dial(SIP/${EXTEN:1}@89.202.174.188)
```

[mobils]

```
exten => _06XXXXXXXX,1,ExecIf($[ "${CALLERID(num)}"="5007"
],Set,CALLERID(all)=Creapolis <5000>)
exten => _06XXXXXXXX,2,ExecIf($[ "${CALLERID(num)}"="5008"
],Set,CALLERID(all)=Creapolis <5000>)
exten => _06XXXXXXXX,3,ExecIf($[ "${CALLERID(num)}"="5009"
],Set,CALLERID(all)=Creapolis <5000>)
exten => _06XXXXXXXX,4,ExecIf($[ "${CALLERID(num)}"="5099"
],Set,CALLERID(all)=Creapolis <5000>)
exten => _06XXXXXXXX,5,ExecIf($[ "${CALLERID(num)}"="5098"
],Set,CALLERID(all)=Creapolis <5000>)
exten => _06XXXXXXXX,6,ExecIf($[ "${CALLERID(num)}"="5097"
],Set,CALLERID(all)=Creapolis <5000>)
exten => _06XXXXXXXX,7,ExecIf($[ "${CALLERID(num)}"="5096"
],Set,CALLERID(all)=Creapolis <5000>)
exten => _06XXXXXXXX,8,ExecIf($[ "${CALLERID(num)}"="5095"
],Set,CALLERID(all)=Creapolis <5000>)
exten => _06XXXXXXXX,9,ExecIf($[ "${CALLERID(num)}"="5094"
],Set,CALLERID(all)=Creapolis <5000>)
exten => _06XXXXXXXX,10,ExecIf($[ "${CALLERID(num)}"="5093"
],Set,CALLERID(all)=Creapolis <5000>)
exten => _06XXXXXXXX,11,ExecIf($[ "${CALLERID(num)}"="5092"
],Set,CALLERID(all)=Creapolis <5000>)
exten => _06XXXXXXXX,12,ExecIf($[ "${CALLERID(num)}"="5091"
],Set,CALLERID(all)=Creapolis <5000>)
exten => _06XXXXXXXX,13,ExecIf($[ "${CALLERID(num)}"="5090"
],Set,CALLERID(all)=Creapolis <5000>)
exten => _06XXXXXXXX,14,ExecIf($[ "${CALLERID(num)}"="5074"
],Set,CALLERID(all)=Creapolis <5000>)
exten => _06XXXXXXXX,15,ExecIf($[ "${CALLERID(num)}"="5073"
],Set,CALLERID(all)=Creapolis <5000>)
exten => _06XXXXXXXX,16,ExecIf($[ "${CALLERID(num)}"="5072"
],Set,CALLERID(all)=Creapolis <5000>)
exten => _06XXXXXXXX,17,ExecIf($[ "${CALLERID(num)}"="5071"
],Set,CALLERID(all)=Creapolis <5000>)
exten => _06XXXXXXXX,18,ExecIf($[ "${CALLERID(num)}"="5070"
],Set,CALLERID(all)=Creapolis <5000>)
exten => _06XXXXXXXX,19,ExecIf($[ "${CALLERID(num):0:1}"!="5"
],Set,CALLERID(all)=Creapolis <5000>)
exten => _06XXXXXXXX,20,Answer()
exten => _06XXXXXXXX,21,Ringing(2)
exten => _06XXXXXXXX,22,Dial(SIP/${EXTEN}@192.169.48.9)
;exten => _06XXXXXXXX,22,Dial(SIP/${EXTEN}@192.169.48.11)
exten => _06XXXXXXXX,23,Congestion()
exten => _06XXXXXXXX,120,Congestion()
```



```
;exten => _6XXXXXXXX,1,Dial(SIP/0${EXTEN}@192.169.48.11)
exten => _6XXXXXXXX,1,Dial(SIP/0${EXTEN}@192.169.48.9)
exten => _9XXXXXXXX,1,Dial(SIP/0${EXTEN}@192.169.48.10)
exten => 0935572395,1,Dial(SIP/935572395)
exten => 0935572394,1,Dial(SIP/935572394)
```

```
;exten => 0653758360,1,Dial(SIP/${EXTEN}@192.169.48.10)
```

```
[atencio]
```

```
exten => _0XXXXX,1,Dial(SIP/${EXTEN}@192.169.48.10)
exten => _0XXXXX,2,Congestion()
exten => _0XXXXX,102,Congestion()
exten => _0XXXX,1,Dial(SIP/${EXTEN}@192.169.48.10)
exten => _0XXXX,2,Congestion()
exten => _0XXXX,102,Congestion()
exten => _0XXX,1,Dial(SIP/${EXTEN}@192.169.48.10)
exten => _0XXX,2,Congestion()
exten => _0XXX,102,Congestion()
```

```
.....
.....
.....Nivells.....
.....
.....
```

```
[nivell1]
```

```
;home
include => home
;include => default
```

```
[nivell2]
```

```
;home i trucades nacionals
include => nivell1
include => nacionals
include => atencio
```

```
[nivell3]
```

```
;home, trucades nacionals i mobils
include => nivell2
include => mobils
include => internacionals
```

```
[nivell4]
```

```
;home, trucades nacionas, mobils i internacionals
include => nivell3
include => internacionals
#include "extensionsexternes.conf"
```

Archivo queues.conf

```
[general]
```

```
persistentmembers = yes
```

autofill = yes
monitor-type = MixMonitor

[operadora]
member=>Agent/1001
member=>Agent/1002
context=home
music=default
strategy=rrmemory

Archivo agents.conf

[general]
persistentagents=yes

[agents]
agent => 1001,4321,Jordi
agent => 1002,4321,Javi

Archivo manager.conf

[general]
displaysystemname = yes
enabled = yes
webenabled = yes
port = 5038
bindaddr = 0.0.0.0

[admin]
secret = 1234
permit=192.168.20.1/255.255.255.0
read = system,call,log,verbose,command,agent,user,config
write = system,call,log,verbose,command,agent,user,config

Archivo SEP de teléfonos IP Cisco

```
<device>  
  <deviceProtocol>SIP</deviceProtocol>  
  <sshUserId>cisco</sshUserId>  
  <sshPassword>cisco</sshPassword>  
  <devicePool>  
    <dateTimeSetting>  
      <dateTemplate>D/M/Ya</dateTemplate>  
      <timeZone>Central Europe Standard/Daylight Time</timeZone>  
    <ntp>  
      <ntp>  
        <name>192.168.6.12</name>
```

```
<ntpMode>Unicast</ntpMode>
</ntp>
</ntp>
</dateTimeSetting>
<callManagerGroup>
  <members>
    <member priority="0">
      <callManager>
        <ports>
          <ethernetPhonePort>2000</ethernetPhonePort>
          <sipPort>5060</sipPort>
          <securedSipPort>5061</securedSipPort>
        </ports>
        <processNodeName>192.168.6.12</processNodeName>
      </callManager>
    </member>
  </members>
</callManagerGroup>
</devicePool>
<sipProfile>
  <sipProxies>
    <backupProxy>192.168.7.13</backupProxy>
    <backupProxyPort>5060</backupProxyPort>
    <emergencyProxy></emergencyProxy>
    <emergencyProxyPort></emergencyProxyPort>
    <outboundProxy></outboundProxy>
    <outboundProxyPort></outboundProxyPort>
    <registerWithProxy>true</registerWithProxy>
  </sipProxies>
  <sipCallFeatures>
    <cnfJoinEnabled>true</cnfJoinEnabled>
    <callForwardURI>x--serviceuri-cfwdall</callForwardURI>
    <callPickupURI>x-cisco-serviceuri-pickup</callPickupURI>
    <callPickupListURI>x-cisco-serviceuri-opickup</callPickupListURI>
    <callPickupGroupURI>x-cisco-serviceuri-gpickup</callPickupGroupURI>
    <meetMeServiceURI>x-cisco-serviceuri-meetme</meetMeServiceURI>
    <abbreviatedDialURI>x-cisco-serviceuri-abbrdial</abbreviatedDialURI>
    <rfc2543Hold>false</rfc2543Hold>
    <callHoldRingback>2</callHoldRingback>
    <localCfwdEnable>true</localCfwdEnable>
    <semiAttendedTransfer>true</semiAttendedTransfer>
    <anonymousCallBlock>2</anonymousCallBlock>
    <callerIdBlocking>2</callerIdBlocking>
    <dndControl>1</dndControl>
    <remoteCcEnable>true</remoteCcEnable>
  </sipCallFeatures>
  <sipStack>
    <sipInviteRetx>6</sipInviteRetx>
    <sipRetx>10</sipRetx>
    <timerInviteExpires>180</timerInviteExpires>
```

```

<timerRegisterExpires>3600</timerRegisterExpires>
<timerRegisterDelta>5</timerRegisterDelta>
<timerKeepAliveExpires>120</timerKeepAliveExpires>
<timerSubscribeExpires>120</timerSubscribeExpires>
<timerSubscribeDelta>5</timerSubscribeDelta>
<timerT1>500</timerT1>
<timerT2>4000</timerT2>
<maxRedirects>70</maxRedirects>
<remotePartyID>true</remotePartyID>
<userInfo>None</userInfo>
</sipStack>
<autoAnswerTimer>1</autoAnswerTimer>
<autoAnswerAltBehavior>false</autoAnswerAltBehavior>
<autoAnswerOverride>true</autoAnswerOverride>
<transferOnhookEnabled>false</transferOnhookEnabled>
<enableVad>false</enableVad>
<preferredCodec>g729a</preferredCodec>
<dtmfAvtPayload>101</dtmfAvtPayload>
<dtmfDbLevel>3</dtmfDbLevel>
<dtmfOutOfBand>avt</dtmfOutOfBand>
<alwaysUsePrimeLine>false</alwaysUsePrimeLine>
<alwaysUsePrimeLineVoiceMail>false</alwaysUsePrimeLineVoiceMail>
<kpml>3</kpml>
<natEnabled>0</natEnabled>
<natAddress></natAddress>
<phoneLabel>CREAPOLIS</phoneLabel>
<stutterMsgWaiting>1</stutterMsgWaiting>
<callStats>true</callStats>

<silentPeriodBetweenCallWaitingBursts>10</silentPeriodBetweenCallWaitingBursts>
<disableLocalSpeedDialConfig>false</disableLocalSpeedDialConfig>
<startMediaPort>16384</startMediaPort>
<stopMediaPort>32766</stopMediaPort>
<sipLines>
  <line button="1">
    <featureID>9</featureID>
    <featureLabel>5705</featureLabel>
    <proxy>192.168.6.12</proxy>
    <port>5060</port>
    <name>5705</name>
    <displayName>5705</displayName>
    <autoAnswer>
      <autoAnswerEnabled>2</autoAnswerEnabled>
    </autoAnswer>
    <callWaiting>3</callWaiting>
    <authName>5705</authName>
    <authPassword>1234</authPassword>
    <sharedLine>false</sharedLine>
    <messageWaitingLampPolicy>1</messageWaitingLampPolicy>
  </line>
</sipLines>

```

```
<messagesNumber>5999</messagesNumber>
<ringSettingIdle>4</ringSettingIdle>
<ringSettingActive>5</ringSettingActive>
<contact>5705</contact>
<forwardCallInfoDisplay>
  <callerName>true</callerName>
  <callerNumber>true</callerNumber>
  <redirectedNumber>false</redirectedNumber>
  <dialedNumber>true</dialedNumber>
</forwardCallInfoDisplay>
</line>
</sipLines>
<voipControlPort>5060</voipControlPort>
<dscpForAudio>184</dscpForAudio>
<ringSettingBusyStationPolicy>0</ringSettingBusyStationPolicy>
<dialTemplate>dialplan.xml</dialTemplate>
</sipProfile>
<commonProfile>
  <phonePassword></phonePassword>
  <backgroundImageAccess>true</backgroundImageAccess>
  <callLogBlfEnabled>1</callLogBlfEnabled>
</commonProfile>
<vendorConfig>
  <disableSpeaker>false</disableSpeaker>
  <disableSpeakerAndHeadset>false</disableSpeakerAndHeadset>
  <pcPort>0</pcPort>
  <settingsAccess>1</settingsAccess>
  <garp>0</garp>
  <voiceVlanAccess>0</voiceVlanAccess>
  <videoCapability>0</videoCapability>
  <autoSelectLineEnable>0</autoSelectLineEnable>
  <webAccess>0</webAccess>
  <spanToPCPort>1</spanToPCPort>
  <loggingDisplay>1</loggingDisplay>
  <loadServer></loadServer>
</vendorConfig>
<versionStamp>1143565489-a3cbf294-7526-4c29-8791-
c4fce4ce4c37</versionStamp>
<networkLocale>US</networkLocale>
<networkLocaleInfo>
  <name>US</name>
  <version>5.0(2)</version>
</networkLocaleInfo>
<deviceSecurityMode>1</deviceSecurityMode>
<authenticationURL></authenticationURL>
<directoryURL></directoryURL>
<idleURL></idleURL>
<informationURL></informationURL>
<messagesURL></messagesURL>
<proxyServerURL></proxyServerURL>
```

```
<servicesURL></servicesURL>
<dscpForSCCPPhoneConfig>96</dscpForSCCPPhoneConfig>
<dscpForSCCPPhoneServices>0</dscpForSCCPPhoneServices>
<dscpForCm2Dvce>96</dscpForCm2Dvce>
<transportLayerProtocol>4</transportLayerProtocol>
<capfAuthMode>0</capfAuthMode>
<capfList>
  <capf>
    <phonePort>3804</phonePort>
  </capf>
</capfList>
<certHash></certHash>
<encrConfig>false</encrConfig>
</device>
```